

Thèse de doctorat

pour l'obtention du grade de

Docteur de l'Université Paris-Est

Spécialité Informatique Linguistique

au titre de l'École Doctorale MSTIC

Présentée et soutenue publiquement par

Elsa Tolone

le 31 mars 2011

Analyse syntaxique à l'aide des tables du Lexique-Grammaire du français

Devant le jury composé par :

Rapporteurs :	Laurence Danlos	(Université Denis Diderot - Paris 7)
	Laura Kallmeyer	(Universität Düsseldorf)
Examinateurs :	Éric de La Clergerie	(INRIA Paris–Rocquencourt)
	Denys Duchier	(Université d'Orléans)
Directeur de thèse :	Éric Laporte	(Université Paris-Est)
Co-directeur de thèse :	Matthieu Constant	(Université Paris-Est)

Remerciements

Avant tout, je tiens à remercier les membres du jury, Laurence Danlos, Laura Kallmeyer, Éric de la Clergerie et Denys Duchier, d'avoir accepté d'évaluer mon travail et pour leur commentaires enrichissants.

Je remercie également mon directeur de thèse, Eric Laporte, pour toutes ses précieuses remarques, notamment lors de la rédaction de mon travail.

Sans oublier mon co-directeur, Matthieu Constant, pour m'avoir conseillé durant mon travail et son aide pour certains outils.

L'aide de Christian Leclère a été pour moi d'une importance capitale puisqu'il est l'auteur principal des tables de verbes et qu'il a bien voulu m'accorder le temps nécessaire pour démystifier toutes mes interrogations.

Ma collaboration avec Stavroula Voyatzi m'a permis également d'aborder les expressions figées et les adverbes avec beaucoup plus de facilité.

Enfin, Benoît Sagot m'a énormément aidé quand à la comparaison des deux formats de lexiques et pour les outils de conversion.

Un grand merci à Éric de la Clergerie pour sa réactivité et l'adaptation de son analyseur syntaxique à mon lexique.

Certains doctorants ont également jouer un rôle important :

- Kyriaki Ioannidou avec son optimisme exemplaire et ses questions acharnées qui m'ont permis d'envisager la poursuite de ce travail pour le grec,
- Wankawee Puangkor avec sa gentillesse et son aide à me dépatauger parmi la masse de fiches cartonnées,
- Myriam Rakho et Antony Sigogne avec l'utilisation du fruit de mon travail dans d'autres applications,
- enfin, aussi bien les doctorants de Paris-Est, de Paris 7, de Nancy, que de la FaMAF m'ont permis de partager de bons moments lors de ces dernières années.

Merci à Noémi Boubel et son équipe du CENTAL, ainsi qu'à Paul Sabatier et son équipe du LIF pour leur accueil chaleureux et leur intérêt lors de mes séminaires.

Je souhaiterais également remercier Tita Kyriacopoulou, qui a su me donner goût à la linguistique, lorsqu'en licence j'ai suivi son cours tout à fait par hasard, et qui grâce à son dynamisme et à ses discussions passionnantes, m'a aidée à choisir ma voie.

Pour finir, je souhaiterais remercier tous mes relecteurs que j'ai déjà cités pour la plupart : j'ajouterais Lidia Varga, ainsi que mes parents, qui ont tous les deux eu le courage de lire ma thèse jusqu'au bout.

Enfin, merci à toute ma famille et à tous mes amis pour m'avoir soutenue durant toutes ces années.

Résumé

Les tables du Lexique-Grammaire, dont le développement a été initié par Gross (1975), constituent un lexique syntaxique très riche pour le français. Elles couvrent diverses catégories lexicales telles que les verbes, les noms, les adjectifs et les adverbes. Cette base de données linguistiques n'est cependant pas directement exploitable informatiquement car elle est incomplète et manque de cohérence.

Chaque table regroupe un certain nombre d'entrées jugées similaires car elles acceptent des propriétés communes. Ces propriétés ont pour particularité de ne pas être codées dans les tables mêmes mais uniquement décrites dans la littérature. Pour rendre ces tables exploitables, il faut expliciter les propriétés intervenant dans chacune d'entre elles. De plus, un grand nombre de ces propriétés doivent être renommées dans un souci de cohérence.

Notre objectif est d'adapter les tables pour les rendre utilisables dans diverses applications de Traitement Automatique des Langues (TAL), notamment l'analyse syntaxique. Nous expliquons les problèmes rencontrés et les méthodes adoptées pour permettre leur intégration dans un analyseur syntaxique.

Nous proposons *LGExtract*, un outil générique pour générer un lexique syntaxique pour le TAL à partir des tables du Lexique-Grammaire. Il est relié à une table globale dans laquelle nous avons ajouté les propriétés manquantes et un unique script d'extraction incluant toutes les opérations liées à chaque propriété devant être effectuées pour toutes les tables. Nous présentons également *LGLex*, le nouveau lexique syntaxique généré des verbes, des noms prédicatifs, des expressions figées et des adverbes.

Ensuite, nous montrons comment nous avons converti les verbes et les noms prédicatifs de ce lexique au format Alexina, qui est celui du lexique Le*fff* (Lexique des Formes Fléchies du Français) (Sagot, 2010), un lexique morphologique et syntaxique à large couverture et librement disponible pour le français. Ceci permet son intégration dans l'analyseur syntaxique FRMG (*French MetaGrammar*) (Thomasset et de La Clergerie, 2005), un analyseur profond à large couverture pour le français, basé sur les grammaires d'arbres adjoints (TAG), reposant habituellement sur le Le*fff*.

Cette étape de conversion consiste à extraire l'information syntaxique codée dans les tables du Lexique-Grammaire. Nous présentons les fondements linguistiques de ce processus de conversion et le lexique obtenu. Nous évaluons l'analyseur syntaxique FRMG sur le corpus de référence de la campagne d'évaluation d'analyseurs du français Passage (Produire des Annotations Syntaxiques à Grande Échelle) (Hamon *et al.*, 2008), en comparant sa version basée sur le Le*fff* avec notre version reposant sur les tables du Lexique-Grammaire converties.

Abstract

Lexicon-Grammar tables, whose development was initiated by Gross (1975), are a very rich syntactic lexicon for the French language. They cover various lexical categories such as verbs, nouns, adjectives and adverbs. This linguistic database is nevertheless not directly usable by computer programs, as it is incomplete and lacks consistency.

Tables are defined on the basis of features which are not explicitly recorded in the lexicon. These features are only described in literature. To use these tables, we must make explicit the essential features appearing in each one of them. In addition, many features must be renamed for consistency sake.

Our aim is to adapt the tables, so as to make them usable in various Natural Language Processing (NLP) applications, in particular parsing. We describe the problems we encountered and the approaches we followed to enable their integration into a parser.

We propose *LGExtract*, a generic tool for generating a syntactic lexicon for NLP from the Lexicon-Grammar tables. It relies on a global table in which we added the missing features and on a single extraction script including all operations related to each property to be performed for all tables. We also present *LGLex*, the new generated lexicon of French verbs, predicative nouns, frozen expressions and adverbs.

Then, we describe how we converted the verbs and predicatives nouns of this lexicon into the Alexina framework, that is the one of the Le*fff* lexicon (*Lexique des Formes Fléchies du Français*) (Sagot, 2010), a freely available and large-coverage morphological and syntactic lexicon for French. This enables its integration in the FRMG parser (French MetaGrammar) (Thomasset et de La Clergerie, 2005), a large-coverage deep parser for French, based on Tree-Adjoining Grammars (TAG), that usually relies on the Le*fff*.

This conversion step consists in extracting the syntactic information encoded in Lexicon-Grammar tables. We describe the linguistic basis of this conversion process, and the resulting lexicon. We evaluate the FRMG parser on the reference corpus of the evaluation campaign for French parsers Passage (*Produire des Annotations Syntaxiques à Grande Échelle*) (Hamon *et al.*, 2008), by comparing its Le*fff*-based version to our version relying on the converted Lexicon-Grammar tables.

Table des matières

1 Introduction

1.1 Contexte

L'analyse syntaxique de textes à l'aide de lexiques syntaxiques est au centre de projets de recherche récents sur le français :

- *Mosaïque* (2006-2007, INRIA) sur le développement de modèles syntaxiques de haut niveau, en considérant, en particulier, la notion de méta-grammaire ;
- *EVALDA-EASy* (2003-2005, Technolangue) sur la mise en place d'une campagne d'évaluation d'analyseurs syntaxiques ;
- *LexSynt* (2005-2007, Institut de la Langue Française) sur le développement de modèles linguistiques pour les lexiques syntaxiques et sémantiques, ainsi que sur la constitution d'un lexique de référence pour le français ;
- *Passage* (2007-2009, ANR) sur la construction automatique d'un très gros corpus annoté en syntaxe par l'utilisation et l'évaluation de plusieurs analyseurs syntaxiques, ainsi que sur l'acquisition d'informations linguistiques et extra-linguistiques à partir de ce corpus ;
- *Séquoïa* (2009-2011, ANR) sur les analyseurs syntaxiques probabilistes à large couverture pour le français ;
- *EDyLex* (2010-2012, ANR) sur l'acquisition dynamique de nouvelles entrées dans des lexiques existants, utilisés au sein de chaînes d'analyse syntaxique et sémantique complètes.

La construction de lexiques syntaxiques n'existe pas que pour le français, des projets existent depuis les années 1990 pour l'anglais.

Le projet *ComLex Syntax* (Grishman *et al.*, 1994) a pour objectif la création d'un lexique syntaxique à large couverture pour l'anglais. La première version de ComLex Syntax a été diffusée par le Linguistic Data Consortium (LDC) en 1994 et la version

finale en 1997. Il s'est poursuivi par la création de *NomLex* (Macleod *et al.*, 1997), un lexique sur les nominalisations de verbes avec une première version publiée en 1999, sa dernière version datant de 2001.

FrameNet (Baker *et al.*, 1998) est un projet fondé sur la sémantique des cadres. Il a pour objectif de documenter la combinatoire syntaxique et sémantique pour chacun des sens d'une entrée lexicale à travers une annotation manuelle d'exemples choisis dans des corpus sur des critères de représentativité lexicographique. Il a débuté en 1997 et se poursuit actuellement.

Citons pour terminer, *VerbNet* (Kipper *et al.*, 2000), un lexique des classes de verbes anglais, qui est la ressource la plus proche du Lexique-Grammaire. VerbNet regroupe par classe les verbes partageant les mêmes comportements syntaxiques et sémantiques. C'est un prolongement des travaux de Levin (1993). De nouvelles versions sortent régulièrement, la version actuelle datant de 2009.

Les tables du Lexique-Grammaire du français (Gross, 1975) constituent une base de données riche et détaillée contenant des informations lexicales, syntaxiques et sémantiques. Elles couvrent diverses catégories lexicales telles que les verbes, les noms, les adjectifs et les adverbes. Chaque table regroupe les éléments lexicaux d'une catégorie lexicale donnée, partageant certaines propriétés syntaxico-sémantiques. Leur développement a été initié depuis plus de quarante ans par M. Gross, au sein du Laboratoire d'Automatique Documentaire et Linguistique (LADL), puis du Laboratoire d'Informatique Gaspard-Monge (LIGM) de l'Université Paris-Est. Mais elles n'ont pas été créées en un format permettant directement leur utilisation dans un programme informatique tel qu'un analyseur syntaxique.

1.2 Objectif

L'objectif de cette thèse est de permettre aux données linguistiques codées dans les tables du Lexique-Grammaire de servir de base de données lexicales pour diverses applications de Traitement Automatique des Langues (TAL), notamment un analyseur syntaxique automatique du français. Pour cela, nous avons converti les tables au format Alexina, qui est celui du lexique Le*fff*, un lexique syntaxique à large couverture pour le français (Sagot et Danlos, 2007; Sagot, 2010), présenté au chapitre 8. En effet, ce format est directement utilisable dans les applications de TAL de haut niveau, y compris celles qui nécessitent une analyse syntaxique profonde.

Parmi les analyseurs qui prennent en entrée un lexique au format Alexina, nous avons choisi l'analyseur FRMG, un analyseur profond à large couverture pour le français (Thomasset et de La Clergerie, 2005), avec une description grammaticale de haut niveau sous la forme de méta-grammaire. Nous avons donc réalisé l'intégration des tables dans cet analyseur, afin d'évaluer leur impact lors d'une l'analyse au chapitre 10.

Avant d'être converties au format Le*fff*, nous avons converti les tables en un format

d'échange, reposant sur les mêmes concepts linguistiques que ceux qui sont manipulés dans les tables. Nous proposons l'outil générique, nommé *LGExtract*, qui permet de générer un lexique syntaxique pour le TAL à partir des tables du Lexique-Grammaire. Il est relié à une table globale, appelée table des classes, dans laquelle nous avons ajouté les propriétés manquantes, et à un unique script d'extraction incluant toutes les opérations liées à chaque propriété devant être effectuées pour toutes les tables. Nous présentons cet outil, ainsi que le lexique obtenu, appelé *LGLex*, au chapitre 7. Le format du lexique *LGLex* a vocation à décrire les tables avec les concepts manipulés par celles-ci, en un format directement exploitable dans les applications de TAL. L'une des utilisations informatiques possibles est la conversion en un autre format, par exemple au format Le*fff*, ce qui suppose de manipuler d'autres concepts linguistiques, ceux manipulés par le Le*fff*.

Mais en premier lieu, nous avons dû modifier les tables pour les rendre directement exploitables, ce qui est détaillé dans les chapitres 5 et 6. En effet, les propriétés communes à chaque table ont pour particularité de ne pas être codées dans les tables mêmes mais uniquement décrites dans la littérature. Pour rendre ces tables exploitables, il faut expliciter les propriétés intervenant dans chacune d'entre elles. De plus, un grand nombre de ces propriétés doivent être renommées dans un souci de cohérence. Nous expliquons les problèmes rencontrés et les méthodes adoptées lors de cette étape d'homogénéisation. Le but étant que les tables soient entièrement corrigées et complétées afin de former un tout cohérent avec notamment une description précise de chaque propriété syntaxico-sémantique et une définition formelle pour chaque table.

Les tables gardent leur raison d'être, car elles sont dans un format facilement manipulable par des linguistes n'étant pas informaticiens. En effet, le format tabulaire facilite le codage car les entrées sont faciles à identifier visuellement et à comparer.

Les tables d'origine sous format papier constituent la version 1. Après avoir été récupérées sous format électronique (Excel), 60% des tables ont été mises en ligne en 2002, ce qui constitue la version 2. Cette thèse a donné jour à la version 3 en septembre 2008, version qui évolue régulièrement. Elle inclut toutes les modifications effectuées sur les tables disponibles, ainsi que les lexiques auxquels elles ont été convertis (*LGLex* et *LGLex*-Le*fff*). Diffusée le 31 mars 2011, la version 3.3 contient la totalité des tables, avec de plus une documentation complète de toutes les propriétés syntaxico-sémantiques des verbes distributionnels, ainsi que la définition formelle de chaque table et l'arbre de classement des verbes distributionnels.

1.3 Plan de la thèse

Le chapitre 2 **(Les analyseurs syntaxiques et les lexiques syntaxiques)** évoque l'analyse syntaxique, l'évaluation d'analyseurs lors de la campagne Passage, ainsi que les lexiques syntaxiques.

Le chapitre 3 **(Les tables du Lexique-Grammaire)** présente les motivations initiales de la création des tables du Lexique-Grammaire. Des exemples de tables sont

donnés et l'ensemble des notations utilisées sont décrites. Les tables constituent le point de départ de notre travail.

Le chapitre 4 (**Le lexique syntaxique** Le*fff* **et l'analyseur syntaxique** FRMG) décrit le format TAL dans lequel nous avons converti les tables. Il s'agit du format Alexina, qui est celui du lexique syntaxique Le*fff*, utilisé par l'analyseur FRMG, dont nous expliquons en détail le fonctionnement.

Le chapitre 5 (**Amélioration des tables et création des tables des classes**) se penche sur les tables du Lexique-Grammaire, l'intérêt de leur format et les améliorations que nous y avons apportées. Nous mettons en avant le format d'origine des tables, tout en expliquant leur classification. Un inventaire précis des tables est établi, ce qui nous permet de soulever l'origine des problèmes de cohérence qu'elles contiennent. Puis, nous expliquons comment nous avons formalisé et homogénéisé les propriétés syntaxico-sémantiques, en listant les différentes solutions apportées. Enfin, nous définissons ce qu'est une table des classes en expliquant son rôle, et nous détaillons le codage qu'il reste encore à faire.

Dans le chapitre 6 (**Constructions de base pour toutes les catégories**), nous décrivons toutes les constructions de base qui ont été modifiées pour l'ensemble des catégories, à savoir les verbes distributionnels, les noms prédicatifs, les expressions figées et les adverbes.

Le chapitre 7 (**Conversion au format** *LGLex*) présente l'outil *LGExtract*, ainsi que son script associé, permettant de convertir les tables de l'ensemble des catégories au format *LGLex*.

Le chapitre 8 (**Conversion au format** Le*fff*) explique comment nous avons effectué la conversion des verbes et des noms prédicatifs du lexique *LGLex* au format Le*fff*, tout en détaillant les fondements linguistiques et la méthodologie pratique de cette conversion. Enfin, la conversion des expressions figées est abordée également.

Dans le chapitre 9 (**Récapitulatif des travaux effectués**) nous faisons un bilan sur l'avancement des travaux de conversion des tables, ainsi que sur l'évolution de la version 3 des tables. Enfin, nous montrons un extrait de la table 31R dans tous les formats disponibles actuellement.

Le chapitre 10 (**Intégration dans** FRMG **et évaluation**) montre comment le lexique a été couplé avec l'analyseur syntaxique FRMG, dont nous justifierons le choix, et évalue cet analyseur sur le corpus de référence de la campagne d'évaluation Passage, en comparant sa version basée sur le Le*fff* avec notre version reposant sur les entrées des verbes simples et des noms prédicatifs des tables du Lexique-Grammaire converties.

Le chapitre 11 (**Conclusion et perspectives**) dresse un bilan du travail effectué sur les tables du Lexique-Grammaire du français afin de les rendre exploitables dans des analyseurs syntaxiques. Puis, nous évoquons l'extension de ce travail à d'autres langues, telles que le grec, afin de construire un lexique syntaxique à partir des tables du Lexique-Grammaire du grec. Enfin, nous terminons par quelques pistes afin d'améliorer le lexique *LGLex*-Le*fff* du français et son intégration dans FRMG.

Première partie

État de l'art

2

Les analyseurs syntaxiques et les lexiques syntaxiques

Ce chapitre évoque l'analyse syntaxique et l'évaluation d'analyseurs lors de la campagne Passage dans la section 2.1, ainsi que les lexiques syntaxiques dans la section 2.2.

2.1 Analyse syntaxique

Cette section situe l'analyse syntaxique en 2.1.1, puis présente la campagne d'évaluation Passage en 2.1.2, et fait une description succincte de quelques analyseurs syntaxiques du français en 2.1.3.

2.1.1 Introduction

Dans le domaine du Traitement Automatique des Langues (TAL), l'analyse syntaxique constitue un point clé dans un grand nombre de traitements automatiques, tels que la compréhension de texte, l'extraction d'information ou la traduction. Le but d'un analyseur syntaxique est de pouvoir construire la structure syntaxique d'une phrase donnée en entrée. Autrement dit, sa tâche est de déterminer pour chaque terme de la phrase sa fonction syntaxique, ainsi que les relations de dépendance syntaxique des éléments de la phrase, telles que sujet-verbe ou verbe-objet. C'est une tâche difficile, en raison de la complexité et de la richesse de la langue. Depuis l'analyseur syntaxique développé par l'équipe de Z. Harris dans les années 50 (Joshi et Hopely, 1996), de nombreux progrès ont été réalisés.

Pour simplifier, on peut classer les différentes approches en deux catégories :

– les analyseurs symboliques qui utilisent une grammaire et/ou un lexique développés manuellement ;

– les analyseurs probabilistes qui reposent sur un modèle acquis à partir d'un corpus annoté manuellement.

L'analyse syntaxique probabiliste profonde obtient ses meilleurs résultats avec le formalisme des grammaires hors-contextes probabilistes (PCFG, *Probabilistic Context-Free Grammar*). Différentes stratégies d'apprentissage sont utilisées : soit lexicalisées (par exemple, Collins (2003)), soit non lexicalisées (par exemple, Klein et Manning (2003)). Notons, pour le français, que des travaux ultérieurs (Crabbé et Candito, 2008) corroborent en grande partie les résultats obtenus sur l'anglais.

L'approche symbolique, bien que laborieuse puisque les ressources sont développées manuellement, permet de construire une base très riche d'informations linguistiques. Il s'agit notamment de décrire les caractéristiques syntaxiques des mots, même si représenter toutes ces données est difficile. Cela induit des modifications importantes si les bons choix de formalisation ne sont pas réalisés dès le départ. C'est dans ce contexte que nous nous plaçons, l'objectif étant de montrer comment nous avons rendu cohérente et avons complété une base de données lexicales, les tables du Lexique-Grammaire, afin d'en faire une ressource utilisable dans les applications de TAL (Tolone, 2009).

2.1.2 Campagne d'évaluation Passage

La campagne d'évaluation Passage (2007-2009)[1] (Hamon *et al.*, 2008) fait suite aux campagnes GRACE (1994-1998)[2] et EASy (2003-2006)[3] (Paroubek *et al.*, 2006). Elle reprend plusieurs des protocoles d'évaluation de cette dernière campagne, avec des corpus sensiblement différents et plus étendus.

Les principaux objectifs de cette campagne soutenue par l'ANR sont les suivants :

– évaluer les analyseurs du français ;
– améliorer l'exactitude et la robustesse des analyseurs du français sur des corpus à grande échelle (270 millions de mots) ;
– exploiter les annotations syntaxiques résultantes pour créer une ressource linguistique plus riche et plus étendue : un treebank pour le français.

Six types de constituants ont été choisis pour cette campagne : groupe nominal, (GN), noyau verbal (NV), groupe adjectival (GA), groupe adverbial (GR), groupe prépositionnel (GP) et groupe prépositionnel à noyau verbal (PV).

Les relations (dépendances entre mots pleins) à relever sont les suivantes : dépendance sujet-verbe (SUJ-V), dépendance auxiliaire-verbe (AUX-V), objet direct (COD-V), autre complément du verbe (CPL-V), modifieur du verbe (MOD-V), subordonnée (COMP), attribut du sujet ou de l'objet (ATB-SO), modifieur du nom (MOD-N), modifieur de l'adjectif (MOD-A), modifieur de l'adverbe (MOD-R), modifieur de la préposition (MOD-P), coordination (COORD), apposition (APPOS) et juxtaposition (JUXT). La Fig. 2.1 donne un exemple d'annotation d'une phrase littéraire (Paroubek *et al.*, 2008).

Le guide d'annotation utilisé lors des campagnes EASy et Passage est le guide des

1. http://atoll.inria.fr/passage/index.fr.html
2. http://www.limsi.fr/TLP/grace/
3. http://www.technolangue.net/article.php3?id_article=198

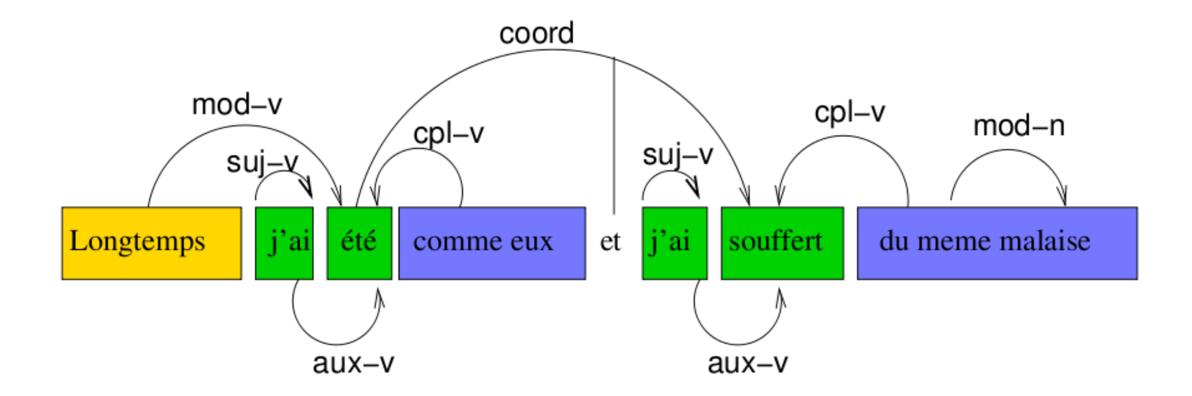


FIGURE 2.1: Annotation d'une phrase littéraire extraite du corpus EASy

annotations syntaxiques de référence PEAS, disponible sur le site http://www.limsi.fr/Individu/anne/Guide/PEAS_reference_annotations_v2.2.html.

Les métriques d'évaluation utilisées sont celles définies à l'occasion de la campagne EASy : la précision, le rappel et la f-mesure, avec différents modes de relâchement de contrainte sur la mesure des frontières de constituants (Paroubek *et al.*, 2006).

Pour la première campagne Passage de 2007, la référence utilisée est un extrait d'environ un million de mots du corpus EASy (Paroubek *et al.*, 2006), de styles variés (journalistique, médical, oral, questions, littéraire, etc.), dont 4 306 phrases constituaient déjà la référence annotée de la campagne EASy en 2004 et 400 nouvelles phrases ont été annotées manuellement depuis. La Tab. 2.1 contient quelques phrases tirées du corpus.

Genre	Exemple de phrase
Journalistique	*Le gouvernement intérimaire a décidé d'asphyxier économiquement le ≪Taylorland≫, en imposant un embargo total sur les marchandises à destination des zones sous contrôle du FNLP.*
Littéraire	*Longtemps j'ai été comme eux, et j'ai souffert du même malaise.*
Médical	*La sensibilité de l'échotomographie pour la définition des calculs vésiculaires de plus de 2mm de diamètre est de 98% environ.*
Parlementaire	*- Monsieur le Président, mes chers collègues, je tiens simplement à faire un rappel au Règlement.*
Mails	*Alors moi je dis chapeau bas pour tes explications mon Jean.*
Oral	*euh l'intervention c'est quoi*

TABLE 2.1: Exemples de phrase par genre dans le corpus EASy

Le corpus Passage pour cette première campagne est composé de :
- Wikipedia (200K mots) : encyclopédique collaboratif libre ;
- Wikinews (18.2K mots) : journalistique libre ;
- Wikilivres (170K mots) : littéraire libre ;
- Europarl (200K mots) : débat Parlementaire européen ;
- JRC-Acquis (120K mots) : juridique européen ;

- Ester (100K mots) : oral transcrit ;
- *Le Monde* (100K mots) : journalistique.
- corpus EASy (1M mots) : multi-styles :
  - *Le Monde* (86K mots) : journalistique ;
  - Parlementaire (82K mots) ;
  - Littéraire (230K mots) ;
  - DELIC (9K mots) : oral transcrit ;
  - Ester (12K mots) : oral transcrit ;
  - Médical (50K mots) ;
  - Questions (52K mots) ;
  - Web (17K mots) ;
  - Mails (150K mots).

Les analyseurs syntaxiques ayant participés à la campagne d'évaluation Passage sont les suivants :

- FRMG, un analyseur syntaxique hybride TIG/TAG dérivé à partir d'une méta-grammaire, développé à l'INRIA (de La Clergerie, 2005b; Thomasset et de La Clergerie, 2005) ;
- SxLFG, un analyseur syntaxique basé sur LFG, développé à L'INRIA (Boullier et Sagot, 2005) ;
- LLP2, un analyseur syntaxique TAG aussi dérivé à partir d'une méta-grammaire, développé au LORIA (Roussanaly *et al.*, 2005) ;
- LIMA, un analyseur syntaxique basé sur des systèmes de règles, développé au LIC2M / CEA-LIST (de Chalendar et Nouvel, 2009) ;
- TagParser, un analyseur syntaxique hybride probabiliste/symbolique, développé par Gil Francopoulo à TAGMATICA (Francopoulo, 2008) ;
- ShP1 et StP1, deux analyseurs syntaxiques basés sur les grammaires de propriétés, développés au LPL et utilisant les satisfactions de contraintes. Le premier est symbolique et déterministe, alors que le second est probabiliste et entraîné grâce aux résultats de l'analyseur syntaxique pendant la campagne EASy (Vanrullen *et al.*, 2006) ;
- Cordial, un analyseur syntaxique commercial basé sur des systèmes de règles, développé par Synapse (Laurent *et al.*, 2009) ;
- SYGMART, un analyseur syntaxique développé au LIRMM (Béchet, 2009) ;
- XIP, un analyseur syntaxique commercial basé sur des cascades de règles, développé par Xerox (Aït-Mokhtar *et al.*, 2002).

Lors de la première évaluation de la campagne Passage qui s'est déroulée fin 2007 [4], 10 analyseurs ont fourni leurs résultats en constituants et seulement 7 ont pu être évalués sur les relations.

Pour l'annotation en constituants, la f-mesure est supérieure à 90% pour la plupart

4. Les résultats de la seconde campagne d'évaluation qui s'est déroulée fin 2009 ne sont pas encore disponibles.

des systèmes et seulement celle de trois systèmes est entre 80% et 90%. La tendance est à peu près la même pour la précision et le rappel. En ce qui concerne les relations, les performances sont moins bonnes que pour les constituants, et la différence entre les différents systèmes est supérieure, étant donné que la tâche est plus complexe. Aucun système n'a une performance au-dessus de 70% en f-mesure, trois sont en-dessous de 60% et deux en-dessous de 50%. Les deux derniers systèmes sont en-dessous de 40%.

2.1.3 Quelques analyseurs syntaxiques pour le français

Détaillons à présent les différents analyseurs syntaxiques ayant participés à la campagne d'évaluation Passage.

FRMG

FRMG (*FRench MetaGrammar*) (de La Clergerie, 2005b; Thomasset et de La Clergerie, 2005) est un analyseur syntaxique profond pour le français. Une description grammaticale de haut niveau sous forme de méta-grammaire sert de point de départ pour la génération d'une grammaire d'arbres adjoints (TAG, *Tree Adjoining Grammar*) (Joshi *et al.*, 1975; Abeillé, 2002). Cette grammaire est transformée par le système DyALog (de La Clergerie, 2005a; de La Clergerie, 2002) en un analyseur syntaxique.

L'analyseur syntaxique FRMG découlant des phases de compilation de la méta-grammaire FRMG ne peut bien sûr fonctionner seul. Il s'intègre dans une chaîne complète de traitement comprenant, en amont, le lexique syntaxique Le*fff* et les nombreux modules de SxPipe (Sagot et Boullier, 2008) en charge de la segmentation, de la correction orthographique et de la détection des entités nommées.

L'analyseur syntaxique FRMG et le lexique syntaxique Le*fff* seront détaillés dans les sections 4.1 et 4.2.

SXLFG

SxLFG (Boullier et Sagot, 2005) est un analyseur syntaxique qui repose sur une variante du modèle des grammaires lexicales fonctionnelles (LFG, *Lexical-Functional Grammars*). Le moteur de SxLFG est un analyseur hors-contexte général qui traite la grammaire support de la LFG. L'ensemble des analyses qu'il produit est représenté sous la forme d'une forêt partagée. L'évaluation fonctionnelle se fait dans une seconde phase au cours d'un parcours bas-haut de cette forêt. L'entrée de l'analyseur est un automate acyclique de mots transformé par le lexeur en un automate acyclique de lexèmes, qui sont les terminaux de la grammaire hors-contexte (CFG, *Context-Free Grammar*) et les structures fonctionnelles sous-spécifiées associées. Un post-traitement (facultatif) permet alors de désambiguïser.

L'analyse de la grammaire support est réalisée par une évolution de l'analyseur Earley : il prend en entrée des automates acycliques de mots et permet de récupérer les erreurs syntaxiques. Traiter un automate acyclique en entrée ne nécessite pas, d'un point de vue

théorique, des changements considérables à l'algorithme Earley, même aidé d'un guide régulier.

LLP2

LLP2 (Roussanaly *et al.*, 2005) est un analyseur syntaxique profond développé en Java, qui s'appuie sur une grammaire d'arbres adjoints lexicalisés (LTAG, *Lexicalized Tree Adjoining Grammar*) (Joshi *et al.*, 1975). L'algorithme implémenté est celui de l'analyse par connexité. L'intégration d'un module de traitement de structures de traits et d'unification, permet de prendre en compte les traits *top* et *bottom* aux nœud des LTAG. En d'autres termes, LLP2 a la capacité de traiter des grammaires FTAG (*Featured-based TAG*) (Vijay-Shanker, 1987).

Du point de vue des ressources, LLP2 s'inspire de l'architecture XTAG qui distingue le lexique morphologique (permettant d'étiqueter les segments et d'identifier les lemmes correspondants), le lexique syntaxique (qui permet la sélection des arbres par filtrage et leur ancrage) et la grammaire (qui contient les arbres TAG).

Le lexique morphologique est majoritairement construit à partir de MULTEXT (Ide et Véronis, 1994). Le lexique syntaxique est extrait du lexique fourni par L. Clément et utilisé par l'analyseur XLFG (Clément, 2001). Un mécanisme par défaut de sélection des arbres élémentaires sur la base de règles reposant sur les traits morphologiques a été mis en place pour pallier les insuffisances du lexique syntaxique.

La grammaire a été engendrée à l'aide d'une méta-grammaire conçue par Benoît Crabbé (Crabbé, 2005) et compilée avec l'outil XMG développé au LORIA (Duchier *et al.*, 2005).

LIMA

L'analyseur LIMA (*LIC2M Multilingual Analyzer*) (de Chalendar et Nouvel, 2009) est implémenté comme un pipeline de modules indépendants appliqués successivement sur un texte. Il met en œuvre une grammaire de dépendance, en ce sens que les analyses produites sont exclusivement représentées comme des relations de dépendance binaire entre les tokens. L'analyseur comprend, entre autres modules, un segmenteur en tokens reposant sur les signes de ponctuation, un étiqueteur morpho-syntaxique, des extracteurs de dépendances à courte et longue portée fondés sur des automates à états finis définis par des règles contextuelles.

TagParser

TagParser (Francopoulo, 2008) est un analyseur syntaxique pour le français et l'anglais. Une implémentation prototypique a été commencée en espagnol. La totalité du code a été écrit en Java de manière portable et industrielle sur une période de sept ans.

L'analyseur est de type montant. Il enchaîne les principaux modules suivants sous forme d'un pipeline : un segmenteur, un analyseur morphologique, un chunker et un module de calcul des relations syntaxiques. La sortie d'analyse comporte trois types de résultat : les constituants sans enchâssement, les relations syntaxiques et les entités

nommées. La communication d'un module à l'autre respecte les principes du LAF (*Linguistic Annotation Framework*), dans le sens où chaque module ajoute une annotation de type déportée (*stand-off* en anglais) sur la donnée transmise.

ShP1 et StP1

Les deux analyseurs ShP1 et StP1 (Vanrullen *et al.*, 2006) sont inclus dans une chaîne de traitement qui est composée d'un segmenteur par règles, et d'un lexique couvrant du français, qui alimentent l'entrée d'un étiqueteur morpho-syntaxique probabiliste. Ce sont deux analyseurs de surface, l'un symbolique et l'autre stochastique.

L'analyseur symbolique ShP1 est un analyseur déterministe. Il repose sur les Grammaire de Propriétés avec une stratégie de coin gauche. La grammaire utilisée est complète en ce sens qu'elle peut être utilisée indifféremment pour une analyse profonde ou superficielle. La particularité de ShP1 est de s'appuyer sur un sous-ensemble de contraintes de la grammaire (en particulier les propriétés de linéarité et de constituance) pour identifier les coins gauches. La stratégie consiste à repérer à partir des coins gauches la frontière droite du chunk sur la base des autres propriétés.

L'analyseur stochastique StP1, comme leur étiqueteur, est basé sur le modèle des patrons (Blache et Rauzy, 2006), un modèle de Markov caché plus performant que les modèles de type N-grammes.

Cordial

Cordial (CORrecteur D'Imprécisions et Analyseur Lexico-sémantique) (Laurent *et al.*, 2009) est un analyseur syntaxique et sémantique développé par la société Synapse Développement. Il est conçu à l'origine pour la correction orthographique et grammaticale. Il associe à des règles générales et très peu liées aux attributs de genre et de nombre (analyseur à relâchement de contraintes) un ensemble d'outils statistiques, en particulier pour effectuer la désambiguïsation grammaticale.

Développé au début des années 90 mais constamment maintenu et enrichi depuis, Cordial est le fondement de nombreux développements : composants linguistiques de nettoyage automatique de texte, extracteur de mots-clés et de phrases-clés, extracteur de thèmes et de concepts, extracteur de terminologie et d'entités nommées, moteur de question-réponse. Conçu pour la langue française, Cordial est devenu bilingue, l'analyseur syntaxique et de nombreuses ressources ayant été progressivement adaptées pour la langue anglaise.

SYGMART

L'analyseur SYGFRAN (Béchet, 2009) est basé sur le système SYGMART (Système Grammatical de Manipulation Algorithmique et Récursive de Texte). C'est un système transformationnel prenant en entrée une chaîne de caractères et proposant en sortie une structure arborescente. Il se fonde sur les algorithmes de Markov, étendus aux arbres. Il permet d'analyser tout langage dont la grammaire pourrait être écrite sous forme de transducteurs d'arbres. L'analyseur SYGFRAN génère une analyse en constituants d'un

énoncé en français. Il est constitué de plus de 11 000 règles dans le but de produire une analyse morpho-syntaxique de la langue française.

XIP

L'analyseur XIP (*Xerox Incremental Parsing*) (Aït-Mokhtar *et al.*, 2002) est développé par la société Xerox Research Center Europe. Il est basé sur un formalisme qui intègre un certain nombre de mécanismes de description d'analyse robuste profonde et peu profonde. Les grammaires XIP ont été développées pour un certain nombre de langues dont le français et l'anglais, et quelques autres sont en cours de développement en dehors de Xerox (japonais, chinois, allemand, tchèque). Les applications principales incluent la reconnaissance d'entités, la désambiguïsation lexicale et structurelle, la résolution de coréférence et plus globalement l'extraction de connaissances.

2.2 Lexiques syntaxiques

Cette section évoque les lexiques syntaxiques en 2.2.1, puis détaille quelques lexiques du français en 2.2.2, avant de finir par une présentation rapide des principaux lexiques de l'anglais en 2.2.3.

2.2.1 Introduction

Un lexique syntaxique est une ressource qui contient l'information sur le potentiel combinatoire d'un prédicat (par exemple, le verbe *dormir* régit un seul argument, le sujet), mais aussi sur le type de ses arguments (par exemple, l'adjectif *fier* se combine avec un syntagme prépositionnel en *de*). Ces informations varient d'une langue à l'autre, elles sont donc essentielles pour l'apprentissage et l'acquisition des langues. Pour le TAL, les informations sur la structure prédicative sont importantes dans la plupart des applications. Briscoe et Carroll (1993) estiment qu'environ la moitié des erreurs des analyseurs syntaxiques repose sur des informations insuffisantes concernant la structure argumentale, tandis que Carroll et Fang (2004) montrent une amélioration significative de la performance d'un analyseur syntaxique enrichi avec un tel lexique. Ces informations jouent également un rôle essentiel pour la génération automatique (Danlos, 1985), la traduction automatique (hye Han *et al.*, 2000), ou l'extraction d'information (Surdeanu *et al.*, 2003).

Néanmoins, ce type d'informations est toujours difficilement disponible. Traditionnellement, de telles ressources ont été développées par des experts humains, par exemple, le Lexique-Grammaire (Gross, 1975) et le DECFC (Dictionnaire explicatif et Combinatoire) de Mel'čuk *et al.* (1999), ce qui garantit leur bonne qualité, mais elles ne sont pas directement adaptées au traitement automatique. Par contre, les ressources informatisées développées automatiquement en vue des applications de TAL, par exemple Bourigault et Frérot (2005), utilisent des méthodes statistiques, ce qui rend les résultats moins fiables.

2.2.2 Les lexiques syntaxiques du français

Plusieurs ressources lexicales syntaxiques pour le français ont été développées depuis de nombreuses années. Les objectifs de ces lexiques sont de définir, pour chaque lemme [5] donné pouvant être le noyau syntaxique d'une phrase, ses différents emplois et, pour chacun de ces emplois, son (ou ses) cadre(s) de sous-catégorisation spécifiant le nombre et le type de ses arguments, et les informations complémentaires qui s'y rapportent.

Les tables du Lexique-Grammaire

Les tables du Lexique-Grammaire constituent la plus ancienne ressource électronique pour le français. Cette ressource regroupe des informations sur la syntaxe des verbes, noms, adjectifs et adverbes dans un format électronique avec une description systématique (Gross, 1975). L'objectif de cette thèse est de la rendre directement exploitable dans un analyseur syntaxique. Elle contient 76 672 entrées correspondant à 62 128 lemmes distincts couvrant uniquement des prédicats. L'ensemble des tables est téléchargeable sous une licence libre (LGPL-LR) sur le site `http://infolingu.univ-mlv.fr` (Données Linguistiques > Lexique-Grammaire > Téléchargement).

Les tables du Lexique-Grammaire font l'objet du chapitre 3.

Lefff

Le Le*fff* (Lexique des Formes Fléchies du Français) est un lexique morphologique et syntaxique pour le français (Sagot et Danlos, 2007; Sagot, 2010). Au niveau morphologique, il contient 536 375 entrées correspondant à 110 477 lemmes distincts couvrant toutes les catégories. Au niveau syntaxique, 10 273 de ces entrées possèdent un cadre de sous-catégorisation. Il est téléchargeable sous une licence libre (LGPL-LR) sur le site `http://gforge.inria.fr/projects/alexina/`.

Le Le*fff* fait l'objet de la section 4.1.

DICOVALENCE

Dicovalence (van den Eynde et Mertens, 2006), successeur du lexique PROTON (van den Eynde et Mertens, 2003), est un dictionnaire syntaxique de verbes construit manuellement dans le cadre méthodologique de l'Approche Pronominale (Blanche-Benveniste *et al.*, 1987). Il est téléchargeable sous une licence libre (LGPL-LR) sur le site `http://bach.arts.kuleuven.be/dicovalence/`. Dicovalence comporte les cadres de valence [6] de 8 313 entrées verbales, correspondant à 3 738 verbes distincts.

5. Le *lemme* est la forme canonique du mot, le plus souvent au masculin singulier. Il représente un ensemble de mots avec la même racine, la même catégorie lexicale et le même sens.

6. Par cadre de valence on entend traditionnellement le nombre et la nature des compléments valenciels du verbe, y compris le sujet, avec mention de leur fonction syntaxique.

Il s'agit de tous les verbes du Petit Robert, à l'exception de ceux appartenant à la terminologie technique et à des emplois archaïques.

Pour identifier la *valence* d'un prédicat (c'est-à-dire ses dépendants et leurs caractéristiques), l'Approche Pronominale exploite la relation qui existe entre les dépendants dits *lexicalisés* (réalisés sous forme de syntagmes) et les pronoms qui couvre en intention ces lexicalisations possibles. Les pronoms (et les paranoms, cf. ci-dessous), contrairement aux syntagmes, aux fonctions syntaxiques ou aux rôles thématiques, ont deux avantages majeurs :

- tout en étant des éléments de référence minimale, ils sont des éléments purement linguistiques, dénués des propriétés qui rendent difficile l'interprétation de la grammaticalité d'énoncés utilisant des dépendants syntagmatiques,
- ils sont en nombre restreint : leur inventaire est fini.

La valence peut donc être obtenue sans qu'il y ait besoin d'un travail d'interprétation, à l'aide d'une vérification systématique et exhaustive des combinaisons entre les différents pronoms et le prédicat verbal. Les pronoms retenus forment un ensemble plus large que ce qui est usuellement désignée par le terme de pronom : il s'agit des pronoms clitiques, des pronoms personnels pleins et des pronoms dits *suspensifs* (qui regroupent ce que l'on appelle habituellement pronoms interrogatifs et adverbes interrogatifs ou indéfinis, comme *à qui*, *quand*, etc.). Sont également pris en compte les *paranoms*, qui se distinguent des pronoms par leur modifiabilité (*rien* modifié dans *rien d'intéressant*) et l'impossibilité de reprise par un syntagme (**il ne trouve rien, les indices*[7] à comparer à *il les trouve, les indices*).

Les combinaisons entre prédicats et pronoms induisent des paradigmes de portée globale. Certains correspondent à peu près aux traditionnelles fonctions syntaxiques (P0 = *je*, *tu*, *il*, *elle*, *qui*, etc. correspond à la fonction sujet, P1 à la fonction objet direct, P2 à la fonction à-objet ou dative, etc.), d'autres permettent des distinctions plus fines que dans d'autres approches (PQ paradigme de quantité, PM paradigme de manière, etc.). Dicovalence proprement dit se présente comme une liste d'entrées correspondant chacune à un emploi d'un lemme verbal (il y a en moyenne 2,4 entrées par lemme). Sont tout d'abord donnés l'entrée et son type : prédicateur simple, verbe adjoint, verbe auxiliaire, verbe copule, verbe de dispositif, construction résultative (van den Eynde et Mertens, 2006). Suivent alors les différents paradigmes qui dépendent du prédicateur (les termes de valences), avec pour chacun d'eux la liste des pronoms et paranoms qui peuvent en être la réalisation. Sont enfin indiquées certaines propriétés complémentaires, dont les passivations possibles (*passif être*, *se passif* et/ou *se faire passif*).

Les Verbes Français

Les Verbes Français (LVF) est un dictionnaire syntaxico-sémantique de verbes construit manuellement par Dubois et Dubois-Charlier (1997) qui est téléchargeable sous une licence libre (LGPL-LR) sur le site du laboratoire MoDyCo (CNRS & Université

7. Un exemple précédé du signe * désigne une phrase non acceptable (ou éventuellement acceptable dans un emploi différent de celui qui est envisagé).

Paris 10) [8]. À chaque verbe est associé une classe syntaxico-sémantique, c'est-à-dire une classe sémantique définie par la syntaxe. Au total, LVF compte 25 610 entrées verbales, correspondant à 12 310 verbes distincts. Un tiers d'entre eux font l'objet de plusieurs entrées. Il y a 290 verbes qui ont 10 entrées ou plus. Exemple d'un cas extrême : le verbe *passer* comporte 61 entrées.
Pour l'élaborer, les auteurs ont utilisé «les méthodes classiques de la grammaire distributionnelle et transformationnelle» (Dubois et Dubois-Charlier, 1997) (p. III). De fait, les principes de LVF sont proches de ceux du Lexique-Grammaire et il s'inscrit dans la tradition des grammaires de Z. Harris. Pour chaque entrée, une seule construction syntaxique est associée, précisant le nombre et la nature (en particulier prépositionnelle) du sujet et des compléments.

Autres

Nous pouvons également mentionner d'autres ressources telles que :
- **LexValf** (Lexique des Valences verbales du Français) (Salkoff et Valli, 2005), qui se consacre à la description de particularités syntaxiques de constructions des unités lexicales verbales. Son objectif est de mettre en commun manuellement diverses données telles que les tables du Lexique-Grammaire, LVF, le Web attesté et de grands dictionnaires de langue française tels que le Grand Robert, le Petit Robert et le Trésor de la Langue Française. Il contient 975 verbes parmi les plus fréquemment utilisés du français ;
- **DiCo** (Dictionnaire Combinatoire) (Mel'čuk et Polguère, 2006), qui est basé sur les mêmes principes que le DECFC de Mel'čuk *et al.* (1999) et représente une modélisation des liens lexicaux du français. Sa finalité première est de décrire chaque entrée selon deux axes : les dérivations sémantiques (relations sémantiques fortes) qui la lient à d'autres entrées de la langue et les collocations (expressions semi-idiomatiques) qu'elle contrôle. Ce dictionnaire inclut 1 075 entrées et 25 540 liens lexicaux ;
- **VfrLPL**, qui fait partie de **DicoLPL** (Dictionnaire du Laboratoire Parole et Langage) (Vanrullen *et al.*, 2005). C'est un lexique syntaxique des verbes du français qui contient 8 800 entrées verbales correspondant à 6 700 verbes distincts. Il décrit pour chaque entrée ses traits morphologiques, sa forme phonétisée, sa fréquence et le lemme sous-jacent. Les verbes contiennent quelques informations concernant la sous-catégorisation. DicoLPL a été constitué sur la base d'un lexique interne au Laboratoire Parole et Langage, puis complété par croisement de ressources existantes et vérification sur corpus.

Une autre ressource publique de grande ampleur est le **TLFi** (Trésor de la Langue Française Informatisé) (Dendien et Pierrel, 2003), développé par le laboratoire ATILF. Ce dictionnaire, bien que très structuré, a été d'abord conçu comme un dictionnaire grand public (pour consultation humaine). Il ne peut donc constituer en l'état une composante d'un modèle linguistique du français. On peut néanmoins en extraire, automati-

8. Lien vers le site de diffusion sur le site FondamenTAL : `http://www.lif.univ-mrs.fr/IMG/html/FondamenTAL.html`

quement ou semi-automatiquement, une grande quantité d'informations précieuses pour l'élaboration d'un modèle formel.

Des ressources lexicales ont également été acquises semi-automatiquement. C'est le cas notamment de **TreeLex** (Kupść, 2007), acquis automatiquement à partir du corpus arboré de Paris 7 (Abeillé, 2003), avec environ 2 000 lemmes verbaux.

Ou encore automatiquement à partir de corpus bruts, comme **LexSchem** (Messiant, 2010), qui a été acquis à partir du corpus journalistique LM10, constitué des articles de 10 années (1991-2000) du journal *Le Monde*. Il contient 7 239 verbes associés à un cadre de sous-catégorisation, dont 3 123 lemmes verbaux différents. De plus, la complémentarité de l'approche probabiliste et symbolique a été étudiée par (Messiant *et al.*, 2009), en comparant LexSchem (lexique acquis par des méthodes automatiques) avec les tables du Lexique-Grammaire (lexique construit manuellement). Ils montrent que les informations acquises par ces deux méthodes sont bien distinctes et qu'elles peuvent s'enrichir mutuellement.

2.2.3 Les lexiques syntaxiques de l'anglais

VerbNet (Kipper *et al.*, 2000) est un lexique des classes de verbes de l'anglais. C'est un projet mené sous l'impulsion de M. Palmer (à l'université de Pennsylvanie, puis à l'université du Colorado). VerbNet regroupe par classe les verbes partageant les mêmes comportements syntaxiques et sémantiques. C'est un prolongement des travaux de Levin (1993).

Une classe de verbes regroupe plusieurs verbes, et identifie des rôles thématiques avec d'éventuelles contraintes de sélection. Elle décrit plusieurs constructions typiques (des *frames*) des verbes membres. La sémantique de l'action ou de l'événement est également précisée. Des sous-classes permettent de décrire d'éventuelles spécialisations d'une classe. On peut en trouver une description dans Kipper-Schuler (2003).

La version la plus récente (VerbNet 3.1), datant de 2009, distingue 270 classes de verbes qui regroupent 5 879 sens de verbes.

ComLex Syntax (Grishman *et al.*, 1994) est un lexique syntaxique à large couverture pour l'anglais, développé par Proteus Project à l'université de New York. La première version de ComLex Syntax a été diffusée par le Linguistic Data Consortium (LDC) en 1994 et la version finale en 1997. Le lexique inclut les entrées de 21 000 noms, 8 000 adjectifs et 6 000 verbes. Les noms ont 9 traits syntaxiques et 9 compléments possibles, les adjectifs ont 7 traits syntaxiques et 14 compléments, les verbes ont 5 traits syntaxiques et 92 compléments et les adverbes ont 11 positions et 12 traits syntaxiques. Pour les 750 verbes plus fréquents, il y a de plus 4 traits syntaxiques et 32 compléments possibles.

NomLex (*Nominalization Lexicon*) (Macleod *et al.*, 1997) est un lexique sur les nominalisations de verbes de l'anglais développé également par Proteus Project à l'université de New York, sous la direction de C. Macleod. NomLex cherche non seulement à décrire les arguments syntaxiques associés aux noms prédicatifs, mais aussi à les relier aux arguments du verbe dont ils dérivent, décrit dans ComLex Syntax. Le lexique compte 1 025 entrées de plusieurs types de nominalisations lexicales, incluant

plus de 1 000 mots distincts. La première version de NomLex a été publiée en 1999 et la dernière version, librement disponible, date de 2001.

FrameNet (Baker *et al.*, 1998), projet mené à Berkeley à l'initiative de C. Fillmore, est de créer une ressource lexicale pour l'anglais fondée sur la sémantique des cadres (*frame semantics*) et confirmée par des attestations dans des corpus. FrameNet a pour objectif de documenter la combinatoire syntaxique et sémantique (ou valence) pour chacun des sens d'une entrée lexicale à travers une annotation manuelle d'exemples choisis dans des corpus sur des critères de représentativité lexicographique. Les annotations sont ensuite synthétisées dans des tables, qui résument pour chaque mot les cadres avec leurs actants sémantiques et arguments syntaxiques. Les fonctions grammaticales reconnues sont sujet, objet et une troisième catégorie pour les compléments obligatoires et les modifieurs, sans les distinguer.

FrameNet 1.5 compte actuellement 960 cadres sémantiques, 11 600 unités lexicales (dont 6 800 complètement annotées) ainsi que 150 000 phrases d'exemples annotés. La totalité des outils et des données est distribuée librement.

3

Les tables du Lexique-Grammaire

Dans la section 3.2 nous présentons les motivations initiales de la création des tables du Lexique-Grammaire. Des exemples de tables sont donnés et l'ensemble des notations utilisées sont décrites dans la section 3.1. Les tables constituent le point de départ de notre travail.

3.1 Notations du Lexique-Grammaire

De manière générale, les notations utilisées sont celles de Gross (1986b) :

- Les constructions syntaxiques sont représentées par des suites de symboles telles que :
  N0 V N1 Prép N2
  dénotant une suite «sujet - verbe - objet direct - complément prépositionnel» comme par exemple dans la phrase *Paul débat cette question avec Luc.*
  N désigne un argument syntaxique, c'est-à-dire le sujet ou un complément essentiel. Les chiffres à droite des N indiquent leur placement de gauche à droite dans la construction de base :
  - N0 : sujet ;
  - N1 : premier complément ;
  - N2 : deuxième complément, etc.

  La notation Ni est utilisée pour désigner le sujet à l'intérieur d'une complétive, comme dans Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W), qui indique qu'une complétive au subjonctif introduite par *que* et de la forme Ni Vsubj W peut être remplacée par un constituant Ni suivi d'une infinitive Vi-inf W, introduite par la préposition *de* (*Paul empêche que Pierre vienne* = *Paul empêche Pierre de venir*). Les chiffres à droite des autres symboles, tels que Adj, Det, Prép, Loc, C, etc. in-

diquent ce même placement. Par exemple, Prép1 désigne la préposition du premier complément, même si la préposition peut ne pas être numérotée dans les constructions, comme c'est le cas pour les verbes (par exemple, dans N0 V Prép N1). De plus, cette numérotation peut être utilisée pour faire référence à un argument syntaxique de la phrase. Par exemple, le pronom *lui-même* peut être noté lui1-même, ce qui indique que le pronom *lui* est coréférent à l'objet N1 (*Le froid a recroquevillé la plante sur elle-même*).

N peut également représenter un substantif ou un groupe nominal lorsqu'un trait sémantique apparaît à droite du N, ou du chiffre (sauf Nnr qui peut désigner une complétive ou une infinitive). Dans la mesure où l'on se focalise très peu sur les déterminants, les adjectifs et les relatives, cette ambiguïté, loin de présenter des inconvénients, permet de représenter simultanément tout un groupe nominal, ainsi que le substantif tête de ce groupe nominal.

Voici quelques exemples de traits sémantiques figurant dans les constructions :
- N0hum : sujet pris dans la classe des substantifs humains (par exemple, **(L'ennemi+Luc)** *quitte la ville*) ;
- N1pl obl : premier complément obligatoirement au pluriel (par exemple, *La bouteille a éclaté en* **mille morceaux**) ;
- N2pc : deuxième complément pris dans la classe des substantifs parties du corps (par exemple, *Paul joint le pouce avec* **l'index**).

Ces mêmes traits sémantiques peuvent faire l'objet à eux seuls d'une propriété distributionnelle écrite sous la forme N0 =: Nhum, N1 =: Npl obl ou N2 =: Npc. Pour cette dernière, on peut indiquer la coréférence avec un substantif de la même phrase en ajoutant un chiffre à droite du trait sémantique pc. Par exemple, C1 =: Npc0 (C1 désignant le substantif tête du premier complément figé dans une expression figée) est employé pour C1pc de N0, c'est-à-dire C1pc portant obligatoirement sur N0 (par exemple, *Max a la* **tête** *ailleurs*) ;

- Les parenthèses contenant plusieurs éléments séparés par le signe + indiquent un choix possible entre ceux-ci ; la lettre E désigne l'élément vide. Ainsi :
  N0 V (E+N1) : *Jean lit (E+un livre)*
  correspond aux deux structures :
  N0 V : *Jean lit*
  N0 V N1 : *Jean lit un livre* ;
- Un signe + entourant deux chiffres sans parenthèses désigne les arguments concernés, comme par exemple dans N0 V N1 + 2, qui représente une phrase avec un objet direct interprété par métonymie comme les deux objets (*Léa a boutonné un pan de sa robe avec l'autre = Léa a boutonné sa robe*) ;
- Les parenthèses ne contenant pas de signe + permettent de délimiter un argument, comme par exemple dans N0 V (N1 de N1c) = N0 V (N1c) (Prép N1) (*Luc stimule la curiosité de Marie = Luc stimule Marie dans sa curiosité*) ;
- Les deux notations =: et = se différencient par le fait que la première signifie «se spécifie ou se développe en» : elle précise une distribution possible d'un ou plusieurs éléments d'une construction ; alors que le = signifie «est transformationnellement lié à» : il suppose toujours l'existence d'une nouvelle construction (représentée à

droite du signe) par rapport à une déjà connue (représentée à gauche). Par exemple, à N1 = Ppv =: le signifie que l'argument à N1 peut être pronominalisé en *le* (*Paul apprend à lire = Paul l'apprend*) ;

- Les crochets représentent une transformation, telle que [extrap] pour l'extraposition, ou [passif par] (respectivement, [passif de]) pour le passif introduit par la préposition *par* (respectivement, *de*) ;
- La notation => désigne une implication, telle que dans impératif => subj, qui implique que la complétive soit au subjonctif lorsque la phrase exprime un ordre (*Max ordonne que Paul vienne*), ou (Nég, interro) => subj, où la négation ou l'interrogation peuvent entraîner la mise au subjonctif de la subordonnée (*Je crois qu'il viendra / Je ne crois pas qu'il vienne / Crois-tu qu'il vienne ?*) ;
- La notation # signifie «différent de», par exemple Loc # de désigne une préposition locative différente de *de* ;
- La notation 'P' désigne un discours direct, comme dans N0 V à N2 : 'P', où c'est la complétive N1 qui peut prendre la forme d'un discours direct (*Luc répond à Léa qu'il va au cinéma = Paul répond à Marie : «Je vais au cinéma»*), ou 'P', V N0 à N2, où de plus, elle apparaît en tête de phrase (*«Je vais au cinéma», répond Luc à Léa*) ;
- La notation <ENT> représente tous les mots faisant partie de l'entrée et la notation <OPT>, ceux faisant partie d'entrées associées[1].

Traditionnellement (Gross, 1986b), les chiffres sont soit en indice pour numéroter les arguments syntaxiques, soit en exposant pour noter la coréférence, ce qui facilite l'interprétation des intitulés. Par exemple, la construction N0 V Loc N1 V0-inf W est notée N_0 V Loc N_1 V^0-inf W. Nous n'avons pas retenu cette distinction dans les intitulés de propriétés des tables, car il n'existe aucun cas où la distinction entre deux intitulés repose uniquement sur la distinction indice/exposant. De plus, l'interprétation des intitulés repose maintenant sur une documentation précise (voir 5.3.3).

Les symboles utilisés sont :

- Adj : Adjectif ; peut être suivi de permut obl pour indiquer que l'adjectif doit être obligatoirement permuté avec le nom ;
- Adj-ment : Adverbe dérivé d'un adjectif, auquel on a ajouté *-ment* ;
- Adj-n : Nom morphologiquement associé à un adjectif ;
- Adv : Adverbe ; le rôle sémantique de l'adverbe peut être spécifié : Advm pour adverbe de manière, Advp pour adverbe de prix, Advl pour adverbe de lieu, Advt pour adverbe de temps, Advfut pour adverbe de temps futur, Advtd pour adverbe de temps duratif, etc. ;
- AdvPhrase : Adverbe de phrase ;
- autre suivi d'un autre symbole (par exemple, Loc1) : Autres valeurs lexicales possibles de ce symbole, en plus de celles représentées dans les propriétés binaires ;
- Aux : Auxiliaire ;
- C : Substantif figé inclus dans la structure d'un argument figé tel que par exemple,

1. Pour les tables verbales, cette notation est employée pour désigner la colonne contenant l'exemple. De plus, <ENGLISH> représente la traduction du verbe en anglais.

Det1 C1 Adj1 dans la construction N0 faire Det1 C1 Adj1 à N2, à contraster avec les arguments libres, ici N0 et N2 ;
- combien ? : Complément précisant une quantité ou une mesure intéressant le procès, et souvent à déterminant numéral ;
- Conj : Conjonction ; la nature de la conjonction peut être spécifiée : ConjC pour conjonction de coordination et ConjS pour conjonction de subordination ;
- Det : Déterminant (simple ou accompagné d'un modifieur, par exemple *un certain*) ; la nature du déterminant peut être spécifiée : Det1 =: déf pour déterminant défini, Det1 = : indéf pour déterminant indéfini, Dnum pour un déterminant numéral ;
- Det N : Déterminant et prédicat nominal ;
- Detc : Déterminant du complément de nom Nc ;
- dé-V : Verbe dérivé de V par un préfixe négatif ;
- du : Article partitif (*du*+*de la*) ;
- E ou <E> : Absence ou effacement d'un élément ; représente l'élément neutre de la concaténation et sert à marquer la séquence vide (préposition zéro, déterminant zéro, etc.) ;
- GN : Groupe nominal ;
- le : Article défini (*le*+*la*+*l'*) ;
- Loc : Préposition locative, c'est-à-dire introduisant un complément de lieu (*dans*, *sur*, *à*, etc.) ;
- Modif : Tout modifieur (relative, adjectif, complément de nom, épithète, etc.) d'un groupe nominal ; un déterminant suivi d'un modifieur (avec la notation Det-Modif) représente un constituant discontinu formé par le déterminant et le modifieur obligatoire ;
- N : Substantif ou groupe nominal, ou argument syntaxique (sujet ou complément essentiel), comme détaillé au paragraphe précédent ;
  - traits sémantiques possibles (attachés au N) : hum (entité humaine) [2], -hum (entité qui n'est pas une personne ni un animal linguistiquement assimilé à une personne), pc (partie du corps d'une personne), pc obl (obligatoirement une partie du corps, ou, par métonymie, une personne), plur (pluriel), pl obl (pluriel obligatoire ou collectif), abs (entité abstraite), conc (objet concret), nr (substantif dénotant une personne, un objet concret, une entité abstraite, une complétive ou une infinitive), pr (nom propre), monnaie (nom de monnaie, une somme d'argent), mes (nom d'unité de mesure), esprit (esprit d'une personne), idée, texte, mot, chemin (situation statique dans laquelle une personne ou une chose peut effectuer un trajet sur ce chemin), coup, trou, couche (couche d'une substance concrète), zone, transport (moyen de transport), instrument, point, trace (trace ou marque), déformation, mal (maladie), psy (psychologique), nc (non contraint) ;
  - rôles thématiques possibles (séparés de N par un espace) : lieu source (lieu source du référent d'un autre argument), lieu de destination (lieu de destination du référent d'un autre argument), nv-dest (nouvelle destination), mouvement (objet

2. La notation hum obl (obligatoirement entité humaine) a été supprimée comme nous le verrons en 6.2.2.

ou lieu en mouvement), lieu du passage (lieu par lequel passe le référent du sujet), lieu du procès (lieu où se déroule le procès), apparition (apparaissant ou étant créé au cours du procès), disparition (disparaissant au cours du procès), bénéficiaire (bénéficiaire du référent d'un autre argument), détrimentaire (détrimentaire du référent d'un autre argument), matériau (matériau utilisé dans le procès), attache (système d'attache interprété comme un instrument), résultat, actif (personne interprétée comme active), neutre (la phrase dénote un événement datable), statique (la phrase dénote une situation statique), métaphore (la phrase a un sens métaphorique), scénique (locatif) ;

- Nc : Complément de nom ;
- Neg : Adverbe de négation, ou pronom clitique *ne* figé avec le verbe V dans Ppv =: Neg ;
- P : Phrase ou proposition ;
- Ppv : Pronom clitique ou particule préverbale (*me+m'+te+t'+se+s'+le+la+l' +les+lui+nous+vous+leur+en+y*) ; il peut être obligatoirement figé avec le verbe V si figé est mentionné, comme par exemple dans Ppv =: en figé ;
- Poss : Déterminant possessif (*mon+ton+son+ma+ta+sa+mes+tes+ses+notre +votre+leur+nos+vos+leurs*) ; un chiffre peut indiquer à quel argument le déterminant possessif est coréférent, par exemple Poss0 est coréférent au sujet N0 ;
- Prép : Préposition ; Prép-adv désigne sa modification sous une forme adverbiale ;
- Qu P : Complétive sans distinction de contenu, introduite par le pronom *que* ; le mode de la complétive peut être spécifié : Qu Pind pour une complétive à l'indicatif et Qu Psubj pour une complétive au subjonctif ; le *ce* de la complétive peut être indiqué, mais également la locution du type *le fait que* introduisant la complétive notée le fait Qu P ;
- Tc : Temps (éventuellement de l'adverbe) faisant partie de l'infinitive ;
- thèmeN1 : Exemple prototypique de nom qui peut occuper la position de l'objet N1 ;
- tout : Déterminant indéfini dérivé de *tout* (*tout+tous+toute+toutes*) ;
- Tp : Temps (éventuellement de l'adverbe) faisant partie de la principale ;
- trajet : Complément locatif introduit par la préposition *sur* ou *le long de* et interprété comme un lieu de passage ;
- un : Article indéfini (*un+une*) ;
- V : Verbe, défini morphologiquement ;
- V-able, V-ateur, V-eur, ou V-eux : Adjectif déverbal lié à V avec un suffixe *-able*, *-ateur*, *-eur*, ou *-eux* ;
- V-adj : Adjectif déverbal lié à V ;
- V-ant : Adjectif déverbal lié à V avec un suffixe *-ant* ou *-ent* (par exemple, dans N0 être V-ant : *Paul sourit = Paul est souriant*), ou verbe au participe présent dans N1 = (N) (V-ant W) (*J'ai repéré que Paul travaille = J'ai repéré Paul travaillant*) ;
- V-inf W : Verbe à l'infinitif, suivi de toute suite de compléments, y compris vide ; le sujet des infinitives peut être spécifié par un chiffre, par exemple :
  N0 V V0-inf W : *Jean veut manger cela*
  V1-inf W V N1 : *Venir ici ennuie Marie*
  V2-inf W V N1 à N2 : *Faire ceci donne du mal à Paul*

Le sujet peut également être coréférent à un complément de nom de l'objet N1 par exemple et noté N0 =: V1c-inf W : *Se présenter aux élections a germé dans la tête de Paul*
Ou encore être coréférent avec le sujet de la complétive objet N1 et noté N0 =: Vi-inf W : *Être trop gros empêche Luc de passer dans le couloir* ;

- V-n : Substantif de la même famille morphologique que V, -n étant un suffixe nominalisateur ; parfois le suffixe, noté Sfx, est précisé, par exemple dans Sfx = -ment ;
- Vc : Verbe faisant partie de l'infinitive ;
- Vconv : Verbe support converse ;
- Vop : Verbe opérateur, généralement causatif ; la phrase *Paul fait boire Marie* est analysé par application de l'opérateur *Paul fait* à la phrase *Marie boit* ;
- Vsup : Verbe support ;
- Vpp : Verbe au participe passé ;
- W : Suite quelconque, éventuellement nulle, de compléments ; cette notation peut indiquer la conservation des autres compléments éventuels dans une construction.

Les autres symboles sont des valeurs lexicales de verbes, prépositions, pronoms, conjonctions, adverbes ou modifieurs.

3.2 Point de départ : Les tables du Lexique-Grammaire

Une part importante des travaux en syntaxe, se concentre sur l'identification et la formalisation de règles générales s'appliquant à une classe étendue de mots. Précisément, les règles de transformation de Chomsky décrivent des relations systématiques entre les diverses structures syntaxiques.

Mais, comme Chomsky (1965) lui-même le remarquait, ces généralisations ont souvent de fortes contraintes lexicales. Pour chaque mot, la question se pose de savoir si une généralisation donnée s'applique à ce mot. En d'autres termes, la description complète de la syntaxe d'une langue implique non seulement l'identification de règles générales, mais également la détermination de quel mot exige, autorise ou interdit, l'application de quelle règle en particulier. Ce fut l'orientation des recherches effectuées par Gross (1975).

Ce dernier constate l'absence d'exhaustivité dans la plupart des démarches, qui se basent uniquement sur quelques exemples choisis sans se soucier de tester la validité de leurs règles ou hypothèses. Comme l'énonce M. Gross, il faut étudier une langue selon les principes élémentaires suivants :

- les expériences doivent porter sur des quantités de données significatives ;
- elles doivent être reproductibles ;
- les exceptions aux règles doivent être recensées ;
- les modifications du cadre formel doivent être très soigneusement justifiées et demeurer extrêmement rares.

L'objectif étant d'accumuler des données empiriques aussi complètes et détaillées que possible, c'est ce qui a donné naissance à une grande base de données lexicales, syntaxiques et sémantiques : les tables du Lexique-Grammaire.

M. Gross prend comme point de départ l'étude des phrases simples du français. Il reprend ainsi l'idée selon laquelle l'unité minimale de sens est la phrase. Le principe qu'il adopte est donc de répertorier les phrases simples et d'étudier les transformations qu'elles peuvent subir. Les propriétés étudiées pour chacune de ces phrases sont essentiellement des propriétés formelles portant sur la syntaxe plutôt que sur la sémantique, ce qui garantit la reproductivité des tests (Gross, 1975). Toutefois, certaines propriétés sémantiques ont été prises en compte lorsqu'elles pouvaient être testées de façon claire.

Les tables du Lexique-Grammaire du français sont les plus développées, mais des descriptions dans d'autres langues telles que l'italien, le portugais, le grec moderne, et des langues non européennes (coréen, malgache, etc.) sont également disponibles.

Leur développement a été initié dès les années 1970 par Gross (1975), au sein du Laboratoire d'Automatique Documentaire et Linguistique (LADL), puis du Laboratoire d'Informatique Gaspard-Monge (LIGM) de l'Université Paris-Est (Boons *et al.*, 1976a,b; Guillet et Leclère, 1992). Ces informations se présentent sous la forme de *tables*. Chaque table correspond à une *classe* qui regroupe les éléments lexicaux d'une catégorie lexicale donnée (verbes, noms, adjectifs, etc.), partageant certaines propriétés syntaxico-sémantiques.

Chaque classe correspond à une ou plusieurs constructions syntaxiques particulières, et rassemble toutes les entrées qui entrent dans cette ou ces constructions. Par exemple, la table 1 des verbes distributionnels contient tous les verbes qui admettent, en plus d'un sujet, un complément infinitif, mais pas un complément qui soit une complétive (par exemple, le verbe *commencer* dans *Luc commence à lire*).

Une table se présente sous forme de matrice : en lignes, les entrées lexicales de la classe correspondante ; en colonnes, les propriétés syntaxico-sémantiques, qui ne sont pas forcément respectées par tous les éléments de la classe ; à la croisée d'une ligne et d'une colonne le signe + ou − indiquant que l'entrée lexicale décrite par la ligne accepte, ou n'accepte pas la propriété décrite par la colonne.

Une propriété syntaxico-sémantique est soit une information qui concerne directement la construction de base associée à la table, soit une transformation de la construction de base, soit une construction supplémentaire (par exemple, les constructions métaphoriques, qui n'ont pas les mêmes distributions d'arguments).

Par exemple, la propriété N0 V signifie pour un verbe «possibilité d'être la tête d'une construction intransitive avec le syntagme nominal sujet initial» ; la propriété [passif par] signifie «diathèse passive possible».

À titre d'illustration, la Fig. 3.1 montre un extrait de la table 33 des verbes distributionnels[3] qui se construisent avec un argument introduit par la préposition *à*. Cela signifie qu'elle est composée des verbes ayant la propriété N0 V à N1 vraie.

Si un verbe a deux sens distincts, il possède deux entrées lexicales puisque chaque sens n'accepte pas le même ensemble de propriétés. Un des exemples qui figure dans la

3. Les notations utilisées dans les intitulés sont détaillés dans la section 3.1.

N0 =: Nhum	N0 =: N-hum	N0 =: Nnr	<ENT>Ppv	Ppv =: se figé	Ppv =: les figé	Ppv =: Neg	<ENT>V	Neg	N0 V	N0 être V-ant	N0 V de N0pc	N1 =: Nhum	N1 =: N-hum	N1 =: le fait Qu P	Ppv =: lui	Ppv =: y	[extrap]	N0idée V Loc N1esprit	<OPT>
+	−	−	***les***	−	+	−	***lâcher Advm***	−	+	−	−	+	−	−	−	−	−	−	***Max les lâche difficilement à Ida***
+	−	−	<*E*>	−	−	−	*renaître*	−	+	+	−	−	+	−	−	+	−	+	*Max renaît au bonheur de vivre*
+	−	−	***se***	+	−	−	***rendre***	−	+	−	−	+	+	+	−	+	+	−	***Max s'est rendu à mon opinion***
+	−	−	***se***	+	−	−	***rendre***	−	+	−	−	+	−	−	−	−	−	−	***Le caporal s'est rendu à l'ennemi***
+	−	−	<*E*>	−	−	−	*renoncer*	−	−	−	−	+	+	−	−	+	−	−	*Max renonce à son héritage*
+	+	+	***ne***	−	−	+	***revenir***	+	−	−	−	+	−	−	+	−	−	−	*La tête de Luc ne revient pas à Max*

TABLE 3.1: Extrait de la table 33 des verbes distributionnels

table 33 est le verbe *se rendre* :
Le caporal s'est rendu à l'ennemi
Max s'est rendu à mon opinion

On peut voir que *se rendre* (dans le sens d'accepter) possède un complément nominal non humain : la propriété N1 =: N-hum est vraie (codage +), alors qu'elle est fausse (codage −) pour *se rendre* (dans le sens de capituler).

Il y a aussi des propriétés dont les valeurs sont des éléments lexicaux. Un extrait de la documentation (voir annexe E) de ces propriétés est présenté dans la Tab. 3.2. Ainsi, les compléments prépositionnels peuvent nécessiter différentes prépositions qui dépendent du prédicat. Dans la table 1 à la Fig. 3.3, les verbes sont suivis d'une préposition et d'une infinitive (ils acceptent donc la propriété N0 V Prép V0-inf W) : *achever* nécessite la préposition *de*, *s'avancer* nécessite la préposition *à* et *aller* ne nécessite pas de préposition, ce qui est symbolisé par le symbole <*E*>.

Remarquons que le clitique *s'* figé avec le verbe *avancer* (*s'avancer*) est contenu dans la colonne <ENT>Ppv, tout comme le pronom obligatoire *les* du verbe *lâcher*, ou encore le clitique *ne* (marquant la négation) figé avec le verbe *revenir*. Pour simplifier le traitement informatique, nous avons créé les colonnes Ppv =: se figé, Ppv =: les figé, Ppv =: Neg, etc. avec un codage binaire, codage qui doit concorder avec les informations contenues dans la colonne <ENT>Ppv, laquelle n'est gardée que pour des raisons de lisibilité. Dans la colonne <ENT>V figure l'entrée verbale, avec y compris éventuellement un adverbe figé avec le verbe. On peut avoir, par exemple, *Advm*, *Advt*, etc., comme pour *lâcher Advm*, qui oblige la présence d'un adverbe de manière, comme dans *Max les lâche difficilement à Ida* ; ou encore *pas*, *plus*, etc., ce qui contraint l'adverbe négatif. Si un adverbe négatif est obligatoirement présent dans la construction, mais a une valeur lexicale libre (*pas*, *point*, *nullement*, *aucunement*, *aucun*, *nul*, *personne*, *rien*, *guère*, *jamais*, *plus*, *nulle part*), il n'est plus inclus dans l'entrée verbale. En effet, la notation *Nég* correspondant à une valeur lexicale libre a été supprimée de l'entrée et fait à présent l'objet d'une nouvelle colonne binaire intitulée Neg, indiquant la présence obligatoire d'un adverbe de négation, y compris si l'adverbe est contraint et spécifié

Propriétés	Description de la propriété
<ENT>V (lexicale)	Forme de l'entrée verbale V. Si un adverbe est figé avec le verbe, il fait partie de la forme. Si des clitiques sont figés avec le verbe, ils ne font pas partie de la forme. Si un adverbe négatif est obligatoirement présent dans la construction, mais a une valeur lexicale libre (*pas*, *point*, *nullement*, *aucunement*, *aucun*, *nul*, *personne*, *rien*, *guère*, *jamais*, *plus*, *nulle part*), il ne fait pas partie de la forme Exemple : *Ce film dure longtemps :* **durer Advt** *; Luc n'arrête pas d'être dérangé :* **arrêter pas** *; Votre geste ne va pas :* **aller**
<OPT> (lexicale)	Phrase d'exemple illustrant le sens de l'entrée verbale V Exemple : *Max achève de peindre le mur ; Max achève les blessés*
<ENT>Ppv (lexicale)	Clitiques figés avec le verbe V. Les informations données dans cette propriété doivent concorder avec celles données dans les propriétés binaires intitulées Ppv =: en figé, etc. Exemple : *Luc* **n'en** *revient pas de ce culot ; Luc* **se** *réserve pour la nuit*
Ppv =: en figé (binaire) Ppv =: la figé Ppv =: le figé Ppv =: les figé Ppv =: se figé Ppv =: y figé	Le pronom clitique *en/la/le/les/se/y* est figé avec le verbe V Exemple : *Je n'en reviens pas de ce culot ; Luc s'en va* *Fermez-la* *La haine le dispute à la colère* *On va les aligner* *Luc se réserve pour la nuit ; Luc s'en va* *Luc y va ; Luc s'y croit*
Ppv =: Neg (binaire)	Le pronom clitique *ne* est figé avec le verbe V, même en l'absence d'un adverbe de négation Exemple : *Luc n'arrête pas d'être dérangé ; Luc ne saurait dormir ici*
Neg (binaire)	Présence obligatoire d'un adverbe de négation Exemple : *Luc n'arrête pas d'être dérangé*
Prép1 (lexicale) Prép2 Prép3	Prépositions de l'objet N1/N2/N3 Exemple : *Max va jusqu'à exiger des dommages :* **jusqu'à** *; Le verre va tomber :* **<E>** *; Qu'Ida est idiote éclate aux yeux de tous :* **Loc** *Max a accrédité auprès des parents la nouvelle que Luc est mort :* **auprès de** *; Max a encadré dans ce texte que Luc était absent :* **Loc** *; Max a pour preuve de cela qu'il ne s'est pas montré :* **pour+comme** *Max désigne Luc à Léa pour faire ce travail :* **pour** *; Max a reçu de Luc comme garantie qu'il aurait une prime :* **<E>**
autre Loc1 (lexicale)	Prépositions de l'objet locatif Loc N1 autres que celles représentées dans les propriétés binaires intitulées Loc N1 =: à N1, etc. Exemple : *Les convives farandolent* **autour** *de la table ; Les délinquants se recrutent* **(parmi+chez)** *les riches*
autre Loc2 (lexicale)	Prépositions de l'objet locatif Loc N2 autres que celles représentées dans les propriétés binaires intitulées Loc N2 =: à N2 destination, etc. Exemple : *Max a découché* **de chez** *Léa ; Max s'en vient* **chez** *vous*

Table 3.2: Extrait de la documentation des propriétés lexicales

N0 =: Nhum	N0 =: Nnc	<ENT>Ppv	Ppv =: se figé	<ENT>V	Aux =: avoir	Aux =: être	N0 est Vpp W	N0 V	Prép1	N1 =: Qu Pind	N1 =: Qu Psubj	Prép V0-inf W = Ppv	N0 V Prép N1hum	N0 V Prép N1-hum	Prép N1 = Ppv	N0 V dans N1	N0 V N1hum	N0 V N1-hum	<OPT>
+	−	*<E>*	−	***achever***	+	−	−	−	***de***	−	−	−	−	−	−	−	+	+	*Max achève de peindre le mur*
+	+	*<E>*	−	***aller***	−	−	−	−	*<E>*	−	−	−	−	−	−	−	−	−	*Le verre va tomber*
+	−	***s'***	+	***avancer***	−	+	−	−	***à***	−	−	−	−	−	−	−	−	−	*Max s'avance à dire que c'est faux*

TABLE 3.3: Extrait de la table 1 des verbes distributionnels

dans l'entrée verbale. C'est pourquoi dans l'exemple *revenir*, nous avons les deux colonnes Ppv =: Neg et Neg codées +. Ces deux colonnes sont toujours codées + pour les entrées accompagnées du clitique *ne* figé avec le verbe, sauf pour l'entrée *savoir* de la table 1, qui sans adverbe négatif a un autre sens : *Luc ne saurait dormir* différent de *Luc n'arrête pas de dormir.*

Un autre exemple, donné à la Fig. 3.4, montre un extrait de la table FNAN des noms prédicatifs. Ce sont des noms avec argument(s) étudiés avec leur verbe support.

Dans les phrases à verbe support, ce n'est pas le verbe qui remplit la fonction de prédicat de la phrase, mais un nom prédicatif (*Luc monte une* **attaque** *contre le fort*), un adjectif prédicatif (*Luc est* **fidèle** *à ses idées*), etc. La distribution du sujet, et éventuellement des compléments essentiels, dépend de cet élément prédicatif. Un verbe support a la particularité de pouvoir s'effacer :
Je connais un politique qui **est** *fidèle à ses idées*
= *Je connais un politique fidèle à ses idées*
Fidèle à ses idées, il a démissionné

Les noms prédicatifs de la table FNAN ont pour verbe support *faire* et entrent dans la construction N0 faire Det N à N1, où le sujet et l'objet indirect sont obligatoirement des noms humains. Cela signifie qu'elle est composée des noms ayant l'ensemble des propriétés N0 faire Det N à N1, N0 =: Nhum et N1 =: Nhum vraies, et les propriétés N0 =: N-hum et N1 =: N-hum fausses. Cette table reconnaît par exemple, le nom *canular* dans la phrase *Max fait un canular à Marie.*

Un exemple appartenant à la classe des expressions figées est donné à la Fig. 3.5. C'est un extrait de la table C6, laquelle accepte la construction N0 V N1 Prép2 Det2 C2. Cette table reconnaît l'expression figée *avoir sur le cœur*, mais aussi *n'en penser pas moins*, où certains éléments de l'entrée sont vides (ici, Prép2 et Det2).

Enfin, la Fig. 3.6 est un extrait de la table PC des adverbes, regroupant les adverbes qui ont la structure Prép C (sans déterminant et sans modifieur). Cette table reconnaît par exemple, l'adverbe figé *de visu* avec le prédicat type *juger*, qui n'est donné qu'à titre indicatif.

<ENT>N	autre Det	Det =: un	Det =: un-Modif	Det =: du	Det =: des	N0 faire le N de V0-inf W	N0hum faire Det N à N1hum sur ce point
cadeau	*<E>*	+	+	−	+	+	−
calembour	*<E>*	+	+	−	+	−	+
câlin	*<E>*	+	+	−	+	−	−
canular	*<E>*	+	+	−	+	+	+
carambouilles	*<E>*	−	−	−	+	−	−
cardiogramme	*<E>*	+	+	−	+	−	−

TABLE 3.4: Extrait de la table FNAN des noms prédicatifs

N0 =: Nhum	N0 =: N-hum	Ppv =: en figé	Ppv =: Neg	<ENT>Ppv	<ENT>V	N0 V N1	N1 =: Nhum	N1 =: N-hum	N1 =: Qu Pind	Qu Pind = V0-inf W	Qu Pind = Aux V0-inf W	Qu Pind = de V0-inf W	N1 =: Qu Psubj	Qu Psubj = V0-inf W	Qu Psubj = de V0-inf W	N1 =: si P	Qu P = Ppv	N1 =: le fait Qu P	<ENT>Prép2	<ENT>Det2	<ENT>C2	[passif]	C2 =: C2 de Nhum	C2 de Nhum = Ppv-C2 =: lui-C2
+	−	−	−	*<E>*	*admettre*	+	−	+	+	−	+	−	−	−	−	+	+	+	*sans*	*<E>*	*démonstration*	−	−	−
+	−	−	−	*<E>*	***avoir***	−	+	+	+	−	−	−	+	−	+	−	+	+	***sur***	***le***	***cœur***	−	−	−
+	−	+	+	***n'en***	***penser pas***	−	−	−	+	−	+	−	−	−	−	−	−	−	***<E>***	***<E>***	***moins***	−	−	−

TABLE 3.5: Extrait de la table C6 des expressions figées

N0 =: Nhum	N0 =: N-hum	Neg obl	Ppv	Prédicat type	<ENT>Prép	<ENT>C	N0 V W de (E+Dind) (façon+manière) C-a	N0 V W C-a-ment	Conjonction	C-a	C-a-ment
+	−	−	<E>	***juger***	***de***	***visu***	−	−	−	<E>	<E>
+	−	−	<E>	*expérimenter*	*in*	*vitro*	−	−	−	<E>	<E>
+	−	−	<E>	*expérimenter*	*in*	*vivo*	−	−	−	<E>	<E>
+	−	−	<E>	*faire N*	*à*	*volonté*	−	−	−	<E>	<E>

Table 3.6: Extrait de la table PC des adverbes

Actuellement, pour le français, nous disposons de 67 tables (et donc classes) de verbes distributionnels simples, la catégorie la mieux décrite, 78 tables de noms prédicatifs [4] simples et composés, 69 tables d'expressions figées [5] (principalement verbales et adjectivales) et 32 tables d'adverbes (adverbes en *-ment* et locutions adverbiales) [6].

Le nombre d'entrées est d'environ 13 800 pour les verbes distributionnels, 12 600 pour les noms prédicatifs, 39 600 pour les expressions figées et de 10 400 pour les adverbes (plus de détails seront donnés dans la section 6.6).

L'ensemble des tables est téléchargeable sous une licence libre (LGPL-LR) sur le site `http://infolingu.univ-mlv.fr` (Données Linguistiques > Lexique-Grammaire > Téléchargement).

4. Sans compter les 5 tables de L. Pivaut qui viennent d'être numérisées sans avoir été corrigées manuellement et ne sont pas encore disponibles (voir 5.5.2).

5. Sans compter 3 tables de L. Danlos (ZD, ZM et Z) qui n'ont pas été numérisées (voir 5.5.2).

6. Sans compter 3 tables d'adjectifs non prédicatifs simples, et 38 tables d'adjectifs prédicatifs simples qui sont en cours d'élaboration, afin de remplacer les 13 tables de L. Picabia qui n'ont pas été numérisées (voir 5.5.2).

4

Le lexique syntaxique Le*fff* et l'analyseur syntaxique FRMG

Dans ce chapitre, nous décrivons dans la section 4.1 le format TAL dans lequel nous avons converti les tables. Il s'agit du format Alexina, qui est celui du lexique syntaxique Le*fff*, utilisé par l'analyseur FRMG, dont nous expliquons en détail le fonctionnement dans la section 4.2. L'ensemble de ces ressources sont incluses dans la chaîne de traitement ALPAGE (Cabrera, 2008).

4.1 Le lexique syntaxique Le*fff* et le format Alexina

Le Le*fff* (Lexique des Formes Fléchies du Français) est un lexique morphologique et syntaxique à large couverture pour le français (Sagot et Danlos, 2007; Sagot, 2010). Il peut être utilisé directement dans les applications de TAL de haut niveau, y compris celles qui nécessitent une analyse syntaxique profonde. Il contient 536 375 entrées correspondant à 110 477 lemmes distincts couvrant toutes les catégories. Toutes les entrées sont fléchies lors de la compilation du lexique. Il est téléchargeable sous une licence libre (LGPL-LR) sur le site `http://gforge.inria.fr/projects/alexina/`.

Le développement du Le*fff* repose sur une architecture d'acquisition et de modélisation de lexiques morphologiques et syntaxiques, nommée Alexina (Architecture pour les LEXiques INformatiques et leur Acquisition), dans laquelle des lexiques pour d'autres langues sont en cours de développement (espagnol, polonais, slovaque, etc.).

Alexina permet de décrire les informations morphologiques et syntaxiques de manière lisible, complète et efficace. Sa flexibilité permet de représenter un grand nombre de phénomènes à travers un format simple, directement utilisable par les divers formalismes grammaticaux (dont LFG et LTAG) qui nécessitent des informations syntaxiques

détaillées.

La dernière version du Le*fff* est la version 3. Elle fait usage de la notion de redistribution : le lexique *intensionnel*, édité par les développeurs du lexique, associe à chaque entrée un cadre de sous-catégorisation [1] canonique, et liste les redistributions possibles à partir de ce cadre. Le processus de *compilation* du Le*fff* intensionnel en Le*fff* extensionnel fléchit les lemmes et construit les différentes entrées pour ces différentes redistributions.

Malgré les différences de représentation linguistique entre le Le*fff* et le Lexique-Grammaire, nous avons pu convertir la majorité des informations syntaxiques contenues dans le Lexique-Grammaire au format Le*fff*, sans avoir à modifier le format d'origine [2].

À présent, nous listons les sources des informations lexicales, ainsi que les données quantitatives du Le*fff* (4.1.1). Ensuite, nous détaillons le processus de compilation du Le*fff* intensionnel en Le*fff* extensionnel (4.1.2), puis en Le*fff* compilé (4.1.3) [3].

4.1.1 Sources des informations lexicales et données quantitatives du Le*fff*

Les informations lexicales incluses dans le Le*fff* sont originaires de différents travaux :

- acquisition automatique (avec validation manuelle) à l'aide de techniques statistiques appliquées sur des corpus bruts (Clément *et al.*, 2004; Sagot, 2005) ;
- acquisition automatique (avec validation manuelle) d'informations syntaxiques atomiques (Sagot, 2006) (chap. 7) ;
- correction et ajout manuel ou guidé par des techniques automatiques, telles que des statistiques simples sur des corpus étiquetés (Molinero *et al.*, 2009) ou la fouille d'erreurs dans les sorties d'analyseurs syntaxiques (Sagot et de La Clergerie, 2008) ;
- étude linguistique de certains phénomènes et de leur représentation dans d'autres ressources, conversion (d'une partie) de ces ressources dans le format Alexina, et validation manuelle de leur fusion automatique avec le Le*fff* ; les ressources principalement utilisées sont les tables du Lexique-Grammaire (Gross, 1975), DICOVALENCE (van den Eynde et Mertens, 2006) et LVF (Dubois et Dubois-Charlier, 1997). Cela a été appliqué entre autres à des constructions impersonnelles (Sagot et Danlos,

1. La description syntaxique et sémantique du Lexique-Grammaire tient systématiquement compte de plusieurs constructions (la construction de base plus une ou plusieurs autres), alors que dans la littérature le terme de *cadre de sous-catégorisation* fait généralement référence à une classification fondée sur des critères qui concernent une construction unique. Dans le cas du Le*fff*, en général une entrée est associée à un cadre de sous-catégorisation canonique, c'est-à-dire ne prenant pas en compte des transformations telles que la passivation, mais qui regroupe plusieurs constructions. Par exemple, l'effacement d'un argument dans une construction est considéré comme une variante de cette construction et est incluse dans le cadre de sous-catégorisation canonique. Mais il est possible d'associer une entrée à plusieurs cadres, pour cela il suffit d'avoir plusieurs entrées ayant le même identifiant. Dans la suite de la thèse, nous emploierons le terme de cadre de sous-catégorisation tel qu'il est utilisé dans le Le*fff*, c'est-à-dire en y incluant plusieurs constructions, mais sans prendre en compte les transformations.

2. Nous verrons en 8.1.4 les informations que nous avons ajoutées.

3. Comme cela a été expliqué dans Fernández González (2010) pour le Le*ffe* (Lexique des Formes Fléchies de l'Espagnol).

2007), constructions pronominales (Danlos et Sagot, 2008), les adverbes en *-ment* (Sagot et Fort, 2007), plusieurs classes d'expressions figées verbales (Laurence *et al.*, 2006), les verbes en *-iser* et *-fier* (Sagot et Fort, 2009) ;
– enfin, un certain nombre des noms et adjectifs viennent initialement du lexique morphologique Multext pour le français (Veronis, 1998).

Au niveau extensionnel, la version actuelle du Le*fff* (3.0.1) (Sagot, 2010) contient 536 375 entrées correspondant à 110 477 lemmes distincts couvrant toutes les catégories. Les chiffres détaillés figurent dans la Tab. 4.1 [4].

Catégorie	Entrées intensionnelles	Lemmes distincts	Entrées extensionnelles
Verbes	7 107	6 825	361 817
Verbes figés	1 868	1 850	3 295
Noms	37 755	37 530	78 338
Adjectifs	10 504	10 483	34 096
Adverbes	4 019	3 584	4 062
Prépositions	226	225	655
Noms propres	52 482	52 185	52 552
Autres	833	632	1 342

Table 4.1: Données quantitatives du Le*fff*

4.1.2 Du Le*fff* intensionnel au Le*fff* extensionnel

Le Le*fff* est basé sur deux niveaux de représentation :
– Un **niveau intensionnel** qui factorise l'information lexicale, de sorte qu'à chaque lemme soit associé une classe morphologique [5] et une information syntaxique détaillée permettant une gestion rapide et simple. L'information lexicale sous forme intensionnelle est organisée dans différents fichiers *.ilex* en fonction de la catégorie lexicale de leurs lemmes.
– Un **niveau extensionnel**, généré automatiquement en compilant le lexique intensionnel (tous les fichiers *.ilex*), dans lequel on associe chaque forme fléchie [6] avec toute son information morphologique et syntaxique : étiquette morphologique, ou cadre de sous-catégorisation de sa redistribution correspondante, etc. Par ce processus on obtient les fichiers sous la forme extensionnelle *.lex*, correspondant à chacun des fichiers *.ilex* compilés.

Lorsque le lexique intensionnel est compilé en un lexique extensionnel, tous les mots appartenant à la famille de chaque lemme sont construits à l'aide de leur classe

4. La catégorie «Autres» comprend les différentes sortes de conjonctions, les déterminants, les interjections, les signes de ponctuation, les pronoms, les préfixes et suffixes, ainsi que des entrées spéciales pour les entités nommées et les mots inconnus.

5. Une *classe morphologique* permet de construire toute les formes associées au lemme.

6. Par *forme fléchie*, on entend chaque mot résultant de l'application de règles de dérivations sur la racine d'un certain lemme. Dit d'une autre manière, la forme fléchie est le mot tel qu'il apparaît.

morphologique[7]. Les classes morphologiques sont définies dans un format qui couvre la plupart des entrées du lexique. Seuls les lemmes qui se fléchissent d'une manière irrégulière sont décrits manuellement dans un fichier avec extension *.mf*.

Chaque entrée dans le lexique intensionnel est généralement définie par un lemme et une catégorie lexicale. Toutefois, il est possible de trouver plusieurs entrées avec le même lemme et la même catégorie lexicale, mais dans ce cas certaines informations morphologiques et syntaxiques diffèrent. Cela permet de diviser un lemme en différents sens sémantiques impliquant des constructions syntaxiques différentes. Cette distinction est conservée une fois le lexique compilé grâce à une numérotation (*___1*) attachée au lemme qui est différente pour chaque entrée lexicale ayant un sens différent.

Une entrée intensionnelle contient les informations suivantes :

– une **classe morphologique**, qui définit les patrons qui construisent toutes les formes fléchies du lemme ;
– un **poids**, qui désigne la fréquence d'apparition d'un mot dans la langue française (dans le but d'aider l'analyseur syntaxique à privilégier une entrée plutôt qu'une autre) : le poids standard est de *100*, puis de rares entrées ont le poids *200*, *300*, *400* ou *600*, comme par exemple l'auxiliaire *avoir* ;
– une **catégorie lexicale**, choisie parmi un ensemble d'étiquettes possibles. Les catégories lexicales sont divisés en deux types : ouvertes[8] (aussi appelées productives) et fermées[9] (aussi appelées grammaticales) ;
– un **cadre de sous-catégorisation**, qui montre explicitement comment le lemme peut être utilisé dans une construction syntaxique particulière. Cette liste énumère les fonctions syntaxiques des arguments possibles du lemme, et la réalisation possible de chacune de ces fonctions ;
– les possibles **redistributions**, qui définissent comment les cadres de sous-catégorisation de base se transforment pour construire les autres cadres de sous-catégorisation.

Soit par exemple l'entrée intensionnelle suivante :

```
clarifier___1       v-er:std
                    100;Lemma;v;
                    <Suj:cln|scompl|sinf|sn,Obj:(cla|scompl|sn)>
                    cat=v;
                    %actif,%passif,%passif_impersonnel,
                    %se_moyen_impersonnel,%ppp_employé_comme_adj
```

Elle décrit une entrée du lemme verbal *clarifier*, qui est transitive directe. Elle a deux arguments réalisés canoniquement par les fonctions syntaxiques *Suj* et *Obj*, qui

7. Cette compilation se fait avec `Alexina-tools`.

8. Les catégories *ouvertes* sont : adjectifs, adverbes, verbes, noms, etc. Par flexion, dérivation, inclusion de néologismes, il est possible d'ajouter de nouvelles formes.

9. Les catégories *fermées* sont : prépositions, pronoms, conjonctions, etc. Elles ne permettent pas d'ajouter de nouvelles formes.

sont décrites entre les chevrons et dont la deuxième est facultative (symbolisé par les parenthèses). Elle admet les redistributions fonctionnelles (préfixées par le symbole %) actif (la distribution par défaut), passif, passif impersonnel (*Il a été clarifié par Pierre que Luc était le coupable*), se moyen impersonnel (*Il s'est clarifié de nombreuses choses à cette réunion*) et participe passé employé comme adjectif (*Les règles sont clarifiées*). Sa classe morphologique est *v-er:std* (classe standard des verbes du premier groupe), sa catégorie lexicale est *v*, et son poids est de *100* (poids par défaut).

À titre d'exemple, l'entrée extensionnelle pour la forme fléchie *clarifiés* et la redistribution *passif* a la forme simplifiée suivante :

```
clarifiés 100 v [pred="clarifier___1
                <Suj:cln|scompl|sn,Obl2:(par-sn)>",
                @passive,@pers,cat=v,@Kmp]
                clarifier___1 PastParticiple Kmp %passif
```

Les fonctions syntaxiques sont définies dans le Le*fff* par des critères proches de ceux de Dicovalence (van den Eynde et Mertens, 2006), qui reposent sur la substituabilité (en prenant en compte pronoms *et* syntagmes), sur le principe de réalisation unique d'une fonction syntaxique pour un prédicat donné, et sur l'identification de la fonction par un paradigme de pronoms (à l'exception des cas à partage d'arguments, c'est-à-dire les attributs).

L'inventaire des fonctions syntaxiques, ainsi que leurs critères définitoires décrits dans (Sagot et Danlos, 2007; Sagot, 2010) sont les suivants :

- *Suj* pour sujet : la forme clitique est celle d'un clitique nominatif ;
- *Obj* pour objet direct : la forme clitique est celle d'un clitique accusatif, substituable par *ceci/cela*, translaté par passivation lorsque c'est possible ;
- *Objà* pour objet indirect canoniquement introduit par la préposition *à* : substituable par *à+pronom non clitique* mais pas par *ici* ou *là(-bas)*, cliticisation possible à l'aide du clitique datif ou du clitique locatif *y* ;
- *Objde* pour objet indirect introduit par la préposition *de* : cliticisation à l'aide du clitique génitif *en*, non substituable par *d'ici* ou *de là* ;
- *Loc* pour argument locatif : substituable par *ici* ou *là(-bas)*, cliticisation à l'aide du clitique locatif *y* (*Pierre va* **à Paris**) ;
- *Dloc* pour argument délocatif : substituable par *d'ici* ou *de là*, cliticisation à l'aide du clitique génitif *en* (*Pierre vient* **de Paris**) ;
- *Att* pour attribut (du sujet, de l'objet ou de l'*à*-objet) et pseudo-objet (*J'ai acheté ceci* **3 euros**) ;
- *Obl* et *Obl2* pour les autres arguments obliques (non cliticisables) ; *Obl2* est utilisé pour les verbes ayant deux arguments obliques, tel que *plaider auprès de quelqu'un en faveur de quelqu'un d'autre.*

Chaque fonction syntaxique peut être réalisée par différentes réalisations, qui sont de trois types :

- pronom clitique : *cln* pour clitique nominatif (**Il** *donne ce livre à Marie*), *cla* pour clitique accusatif (*Il* **le** *donne à Marie*), *cld* pour clitique datif (*Il* **lui** *donne ce livre*), *y* pour clitique locatif (*Max* **y** *va*), *en* pour clitique génitif (*Max* **en** *mange*) ;
- syntagme direct : *sn* pour syntagme nominal (**La belle dame** *arrive*), *sa* pour syntagme adjectival (*La robe est* **verte**), *sinf* pour syntagme infinitif (*Pierre est parti* **dire aurevoir**), *scompl* pour syntagme phrastique fini (*Pierre dit* **que Marie est belle**), *qcompl* pour interrogative indirecte (*Pierre dit* **combien il gagne**) ;
- syntagme prépositionnel : un syntagme direct précédé d'une préposition, comme *de-sn*, *à-sinf* ou *pour-sa* ; *à-scompl* et *de-scompl* représentent les réalisations en *à/de ce que P*).

Enfin, une fonction dont la réalisation est facultative voit sa liste de réalisations possibles mise entre parenthèses.

Des informations syntaxiques complémentaires (contrôle, mode des complétives, etc.) sont notées par des *macros* (*@CtrlSujObj*, *@ComplSubj*, etc.) dont l'interprétation formalisée dépend du contexte d'utilisation (voir 8.1.4).

4.1.3 Du Le*fff* extensionnel au Le*fff* compilé

Le Le*fff* sous forme extensionnelle doit être compilé par le lexicaliseur *Lexed* pour être utilisé dans les analyseurs syntaxiques basés sur des méta-grammaires [10]. Pour cela, les informations morpho-syntaxiques présentes dans les différents fichiers du Le*fff* extensionnel doivent être rassemblées en un seul fichier, *dico.xlfg*, qui est ensuite compilé par *Lexed* en un automate à états finis dans le fichier *dico.xlfg.fsa*.

FRMG *Lexer* utilise ce Le*fff* compilé pour étiqueter les unités lexicales d'un texte donné en entrée avec les informations morpho-syntaxiques.

Les informations du Le*fff* compilé sont adaptées par FRMG *Lexer* au format de FRMG. Par exemple, la catégorie lexicale adjectif est désignée par un *a* dans le lexique Le*fff*, tandis que FRMG *Parser* note les adjectifs par *adj*. La même chose se produit avec les pronoms (*p* et *pro*) ou des adverbes (*r* et *adv*). En revanche, la catégorie lexicale verbe est désignée par *v* aussi bien dans le Le*fff* compilé que dans l'analyseur syntaxique

L'unité lexicale *promet* est ainsi étiquetée :

```
'C'(1,
    lemma{ lex     => promet,
           truelex => 'E1F2|promet',
   lemma   => promettre,
   cat     => v,
   top     => v{diathesis => active,
                            mode => indicative,
```

10. *Lexed* est inclus dans l'architecture `Alexina-tools` et distribué sous licence GPL : http://www.labri.fr/perso/clement/lexed/

```
                        number => sg,
                        person => 3,
                        tense => present},
   anchor  => tag_anchor{ name =>
                      ht{arg0 => arg{function => suj,
                                  kind => subj,
                                  pcas => (-),
                                  real => cat[cln,'CS','S','N2',
                                              prel,pri,'PP',(-)]},
                         arg1 => arg{function => obj,
                                  kind => kind[obj,prepvcomp,scomp,
                                                              (-)],
                                  pcas => prep[de,(-)]},
                         arg2 => arg{function => 'objà',
                                  kind => kind[prepobj,(-)],
                                  pcas => prep['à',(-)]},
                      ctrsubj => subj,
                      diathesis => active,
                      imp => '-',
                      refl => (-)},
    coanchors => [],
    equations => []
}
```

Ce format, également utilisé par l'analyseur syntaxique FRMG *Parser*, est appelé hypertag. La structure d'un hypertag est la suivante :

- *lex* : le mot (forme) auquel est attribué l'hypertag tel qu'il apparaît dans le Le*fff*.
- *truelex* : le mot tel qu'il apparaît dans le texte d'entrée. Il peut ne pas correspondre à sa forme correspondante du Le*fff* pour cause de contraction orale ou écrite, fautes d'orthographe, etc.
- *lemma* : le lemme de la forme en question.
- *cat* : la catégorie lexicale du mot.
- *top* : des informations plus détaillées à propos de la forme, telles que le genre, le nombre, la personne, le mode, la diathèse, l'auxiliaire requis, etc.
- *anchor* : les informations syntaxiques présentes dans l'entrée extensionnelle faisant référence à des arguments du cadre de sous-catégorisation [11], la redistribution, si elle est ou non impersonnelle (*imp*) et si elle est ou non réflexive (*refl*). Ce sont des informations spécifiques qui servent de liaison entre le lexique et la syntaxe. L'information contenue dans le champ *anchor* de l'hypertag décrit la structure représentant

11. *kind* et *pcas* sont des attributs de chaque argument du cadre de sous-catégorisation : le premier décrit les fonctions syntaxiques qui peuvent avoir un argument particulier du verbe, le second énumère l'ensemble (éventuellement vide) des prépositions pouvant précéder l'argument. Il y a deux attributs supplémentaires, qui ne sont pas présents dans l'exemple : *real* (réalisations possibles des fonctions syntaxiques) et *extracted* (indique si l'argument se trouve dans une une position non canonique).

le mot lors de l'analyse syntaxique.

On retrouve dans un hypertag l'ensemble des informations présentes dans la forme extensionnelle correspondant à l'unité lexicale, restreint aux types de traits supportés par FRMG *Parser*[12].

4.2 L'analyseur syntaxique FRMG

L'ensemble des hypertags fourni par FRMG *Lexer* est l'entrée immédiate de FRMG *Parser*, responsable de la prochaine étape dans la chaîne de traitement linguistique : l'analyse syntaxique. FRMG *Parser* est un analyseur profond à large couverture pour le français. Une description grammaticale de haut niveau, sous la forme de méta-grammaire (appelée également FRMG, tout comme l'analyseur syntaxique), sert de point de départ pour générer une grammaire d'arbres adjoints (TAG, *Tree Adjoining Grammar*) (Joshi *et al.*, 1975; Abeillé, 2002) par MGCOMP (de La Clergerie, 2005b) (4.2.1). Cette grammaire TAG est transformée par l'environnement DyALog (de La Clergerie, 2005a) en un analyseur syntaxique (4.2.2). Enfin, la sortie de l'analyse est traitée par la ressource *Forest utils* (4.2.3)[13]. Rappelons que l'ensemble de ces ressources sont incluses dans la chaîne de traitement ALPAGE (Cabrera, 2008).

4.2.1 FRMG et MGCOMP

Les grammaires TAG sont formées d'arbres partiels d'analyse, appelées arbres élémentaires, et qui sont de deux types : arbres initiaux sur lesquels on peut effectuer l'opération de substitution, et arbres auxiliaires, utilisés pour les insertions répétables, et sur lesquels on peut effectuer l'opération d'adjonction. Un nœud feuille étiqueté par un non terminal peut être substitué par un arbre initial α, comme le montre la Fig. 4.1. Une adjonction insère le contenu d'un arbre auxiliaire γ au niveau d'un nœud X, en découpant γ en deux parties. Le noeud X situé dans la partie supérieure (*top*) de β est remplacé par la racine de γ, le noeud X racine de la partie inférieure (*bottom*) de β est remplacé par le noeud pied de γ. Un exemple est donné à la Fig. 4.2.

Les FTAG (*Feature-Based TAG*) (Vijay-Shanker, 1987) sont une variante usuelle de TAG, dans laquelle les nœuds sont décorés par une paire d'attributs *top* et *bot*, généralement exprimés comme des structures de traits.

Pour couvrir le plus largement possible les constructions syntaxiques d'une langue donnée, il est nécessaire de décrire un maximum de structures élémentaires. La taille des grammaires à large couverture fait que leur conception et leur maintenance est une tâche difficile, nécessitant que celle-ci soit (semi-)automatique. En effet, le domaine de localité propre aux arbres TAG implique une explosion combinatoire du nombre d'arbres, ainsi que de leurs sous-arbres correspondants. Par exemple, la structure d'un arbre verbal se

12. Le Le*fff* dispose d'informations morpho-syntaxiques d'un niveau plus profond que FRMG *Parser* est capable d'utiliser actuellement.

13. Comme cela a été expliqué dans Fernández González (2010) pour SPMG (SPanish MetaGrammar), l'analyseur syntaxique de l'espagnol.

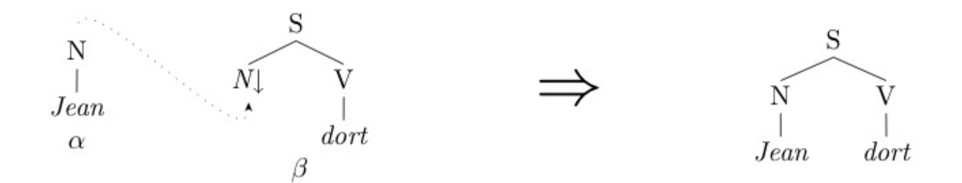


FIGURE 4.1: Exemple de substitution

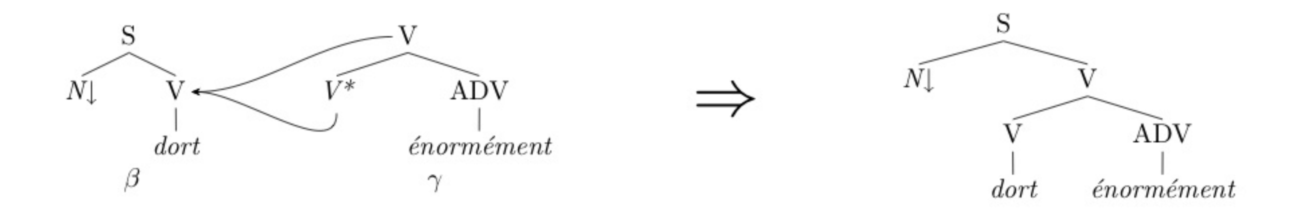


FIGURE 4.2: Exemple d'adjonction

trouve dans tous les arbres ancrés par des verbes. Modifier la description d'un des sous-arbres de l'arbre verbal implique à priori la modification de tous les arbres contenant cet arbre verbal, ce qui entraîne des problèmes de maintenance. Les méta-grammaires (Candito, 1999) fournissent une solution élégante dans ce contexte, en évitant l'explosion combinatoire (Thomasset et de La Clergerie, 2005). Ce type de structures est un outil permettant de faire face aux problèmes apparus dans le développement de larges grammaires TAG (Martin, 2006).

Les méta-grammaires introduisent un haut niveau d'abstraction dans la description des restrictions sur et entre les nœuds qui définissent les structures syntaxiques élémentaires de la langue, en les combinant en classes relativement simples, insérées dans une hiérarchie d'héritage multiple. Parmi ces restrictions, on peut citer le domaine (strict ou immédiat) d'un nœud sur un autre, la précédence linéaire, ainsi que les restrictions de décoration (pour les structures de traits) sur les nœuds ou sur la classe (Thomasset et de La Clergerie, 2005). Cette description grammaticale de haut niveau sert de point de départ pour la génération d'une grammaire TAG pour une langue spécifique. Une méta-grammaire pour le français a été développée (FRMG, *FRench MetaGrammar*), afin d'obtenir une grammaire TAG pour le français. Les méta-grammaires, y compris FRMG, ont les caractéristiques suivantes (Martin, 2006) :

- **Restrictions topologiques** : chaque classe de la hiérarchie contient une description partielle de la structure des arbres TAG élémentaires. Pour cela, on emploie les relations suivantes :
  - (=) égalité : deux identifiants de nœuds reliés par l'opérateur d'égalité sont équivalents, c'est-à-dire les deux identifiants se réfèrent à un même nœud.
  - (<) précédence : indique l'ordre entre deux nœuds frères.
  - (>>) domination immédiate ou directe : un nœud domine directement un autre,

  lorsque le premier est le père du second.
  - (>>+) domination indirecte : un nœud domine indirectement un autre, lorsque le premier est *ancêtre non direct* du deuxième. Un ancêtre non direct signifie que le nœud situé à un niveau supérieur de l'arbre (nœud dominant) n'est pas le père du nœud dominé.
- **Description partielle d'arbre et arbres TAG minimaux** : les arbres décrits sont souvent des *quasi-arbres* (Rogers et Vijay-Shanker, 1992). Un quasi-arbre est une description permettant de construire un nombre infini d'arbres satisfaisant les restrictions.
- **Restrictions de l'unification** : restrictions supplémentaires décrites par des déclarations ou des équations de structures de traits attribuées à certains nœuds ou classes, et sujets à l'unification.
- **Ressources et besoins** : chaque classe peut être consommatrice ou fournisseur d'une ou plusieurs ressources particulières interprétées comme des fonctions syntaxiques (la concordance, par exemple). Chaque ressource peut être consommée directement au niveau de la classe ou par un *espace de noms*. Les espaces de noms permettent à une classe de demander plusieurs fois la même ressource, aussi longtemps que la demande se situe dans des espaces de noms différents pour éviter les conflits. Par exemple, une ressource de concordance en genre et nombre, entre un nœud et son père, sera requise à plusieurs reprises par différentes classes.
- **Gardes** : une classe peut contenir des gardes sur certains nœuds. Une garde est une restriction conditionnelle qui, en fonction de l'existence d'un nœud particulier, conduit à la validation ou le rejet des équations de structures de traits décrites dans la partie droite de la garde. Une garde est exprimée par des *équations de chemins*. Celles-ci sont de la forme :
  `~X => noeud(Y).chemin = valeur(v1),...;`
  pour exprimer le cas de la non existence d'un nœud X, ou
  `X => noeud(Z).chemin = valeur(v2),...;`
  pour exprimer le cas de la présence du même nœud X.
  Un exemple de gardes est présenté ci-dessous.
- **Langage concret de la méta-grammaire** : pour implémenter les classes de la méta-grammaire qui décrivent un langage donné, on utilise un formalisme concret (de La Clergerie, 2005b).

Voici un exemple de la classe représentant les noms communs[14] :

```
class cnoun {
   %% Model for Common nouns
   <: noun;
   N2 >> N;
   N >> Nc;
   N2 >> det;
   det < N;
```

14. L'exemple a été simplifié et réunit les classes _cnoun et cnoun.

```
8:      Nc=Anchor;
9:
10:     node N : [cat: N];
11:     node det : [cat: det, type: subst];
12:     node Nc : [cat: nc];
13:
15:     node(det).top.number = node(N2).bot.number;
15:     node(det).top.gender = node(N2).bot.gender;
16:     node(det).top.wh = node(N2).bot.wh;
17:     node(Anchor).bot.person = value(3);
18:     node(N2).bot.time = node(Nc).top.time;
19:     node(N2).bot.hum = node(Nc).top.hum;
20:
21:     - nc::agreement; Nc = nc::N;
22:     - n::agreement; N = n::N;
23:
24:     det =>
25:         node(N2).bot.sat = value(+);
26:     ~ det =>
27:         node(N2).bot.sat = value(-);
28:         node(N2).bot.wh = value(-),
29: }
```

La ligne 3 indique que la classe *cnoun* hérite de la classe mère *noun*. Les lignes 4 à 7 déclarent les relations topologiques, qui doivent maintenir les nœuds impliqués dans le fragment de l'arbre décrit. Notons que le nœud *det* doit précéder le nœud *N* (ligne 7). L'arbre auquel nous nous référons apparaît à la Fig. 4.3. Le nœud *Nc* est l'ancre de l'arbre (ligne 8).

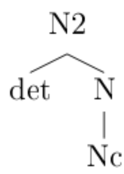


FIGURE 4.3: Structure syntaxique d'un syntagme nominal

Les lignes 10 à 12 déclarent directement les structures de traits pour les nœuds *N*, *det* et *N2*. Les lignes 14 à 19 expriment des restrictions d'unification avec des équations de chemins. Concrètement, dans ces restrictions s'établissent la concordance en genre et nombre, entre les nœuds *det* et *N2*. On vérifie également si les deux nœuds appartiennent à une phrase interrogative (*wh*) et on définit que la personne d'un nom commun est la troisième. De plus, on établit la concordance en temps et trait humain, entre les nœuds *N2* et *Nc*.

Les lignes 21 et 22 indiquent que la classe *cnoun* requiert deux fois la ressource *agreement* (concordance) sur des espaces de noms différents. Cette ressource est fournie par une autre classe de la méta-grammaire qui a un nœud nommé *N*.

La première garde, présente dans les lignes 24 et 25, indique que si le substantif défini par la classe possède un déterminant (*det*), alors le syntagme nominal qui les réunit tous les deux (*N2*) est saturé. La seconde garde, qui occupe les lignes 26 à 28, décrit que si le substantif n'est pas accompagné d'un déterminant, il n'est pas saturé. La négation est écrite avec le signe ∼. De plus, il ne s'agit pas d'un syntagme nominal interrogatif, mais d'un syntagme nominal à l'intérieur d'une phrase interrogative. Cette garde se base sur l'idée qu'un substantif, dans une phrase interrogative, est toujours accompagné d'un déterminant interrogatif. Ainsi, le déterminant *quelle* accompagne le substantif *maison* dans la phrase *Quelle maison est la tienne ?*

La méta-grammaire FRMG est ensuite compilée en une grammaire TAG par le compilateur MGCOMP (Thomasset et de La Clergerie, 2005).

4.2.2 DyALog et FRMG *Parser*

À partir de la grammaire TAG du français et de DyALog (de La Clergerie, 2005a; de La Clergerie, 2002), on obtient un analyseur syntaxique profond, robuste, hybride TAG/TIG et à large couverture pour le français, nommé FRMG *Parser*.

Les grammaires d'insertion d'arbres (TIG, *Tree Insertion Grammar*) (Schabes et Waters, 1995) sont une variante des TAG restreignant les arbres auxiliaires de sorte qu'ils ne puissent s'insérer qu'à droite ou à gauche du nœud d'adjonction. Cette condition implique en particulier que les arbres auxiliaires aient leur dorsale (c'est-à-dire le chemin de la racine au pied) comme frontière gauche ou droite. L'intérêt majeur des grammaires TIG provient du fait qu'elles sont analysables, comme les CFG, avec une complexité en $O(n^3)$ alors que les TAG le sont en $O(n^6)$, où n dénote la longueur de la chaîne d'entrée. De plus, la plupart des grammaires TAG sont essentiellement TIG et il est en fait possible de construire des analyseurs syntaxiques hybrides TAG/TIG (Alonso et Díaz, 2003). DyALog peut analyser une grammaire TAG pour identifier les parties TIG afin de construire de tels analyseurs hybrides TAG/TIG.

À la différence d'autres grammaires TAG à large couverture comprenant plusieurs milliers d'arbres, FRMG *Parser* se caractérise par un très faible nombre d'arbres (moins de 200 arbres). Ceci est réalisé grâce à la capacité descriptive des méta-grammaires et aux mécanismes de factorisation fournis par DyALog (de La Clergerie *et al.*, 2009).

DyALog est un environnement de compilation et d'exécution des analyseurs syntaxiques tabulaires (de La Clergerie, 2005a; Alonso *et al.*, 1999). Une de ces caractéristiques étant la réalisation d'une analyse préliminaire de la grammaire TAG, pour déterminer quels sont les arbres pouvant être compilés en arbres TIG, en raison de leur complexité moindre. Afin de réduire le nombre d'arbres dans la grammaire TAG, DyALog effectue un processus de factorisation en appliquant des opérateurs sur les sous-arbres de la grammaire, comme la disjonction, la fermeture transitive ou l'entrelacement de deux séquences de sous-arbres. Ces opérateurs ne changent pas la nature du forma-

lisme des grammaires, mais ils permettent de réduire de façon exponentielle le nombre d'arbres qui composent la grammaire TAG (de La Clergerie *et al.*, 2009). Cela permet de disposer d'une grammaire beaucoup plus compacte et efficace lors de la construction de l'analyseur syntaxique.

D'autre part, DyALog n'impose pas de restrictions de lexicalisation sur les arbres de la grammaire TAG. Ainsi, certains arbres de la grammaire TAG pour le français ne sont pas ancrés.

La complexité des arbres qui composent la grammaire TAG est illustrée par la Fig. 4.4, qui présente une vue simplifiée d'un arbre verbal canonique pour la voix active. Il s'agit de l'arbre #111, qui résulte du croisement de 25 classes terminales, comprend 43 nœuds et est contrôlé par 35 gardes, où : *S* est la phrase, *VMod* est un modificateur verbal (incluant le sujet de la phrase), *N2* est un syntagme nominal, *CS* est une subordonnée conjonctive, *PP* est un syntagme prépositionnel, *V1* est un syntagme verbal, *Infl* inclut le verbe et un clitique nominatif, *comp* est un attribut, *cln* est un clitique nominatif, *cla* est un clitique accusatif, *cld* est un clitique datif, *prep* est une préposition, *csu* est une conjonction subordonnée, *advneg* est un adverbe de négation, *clr* est un clitique réflexif, *clseq* est une séquence de clitiques, et *v* est le verbe et l'ancre de l'arbre.

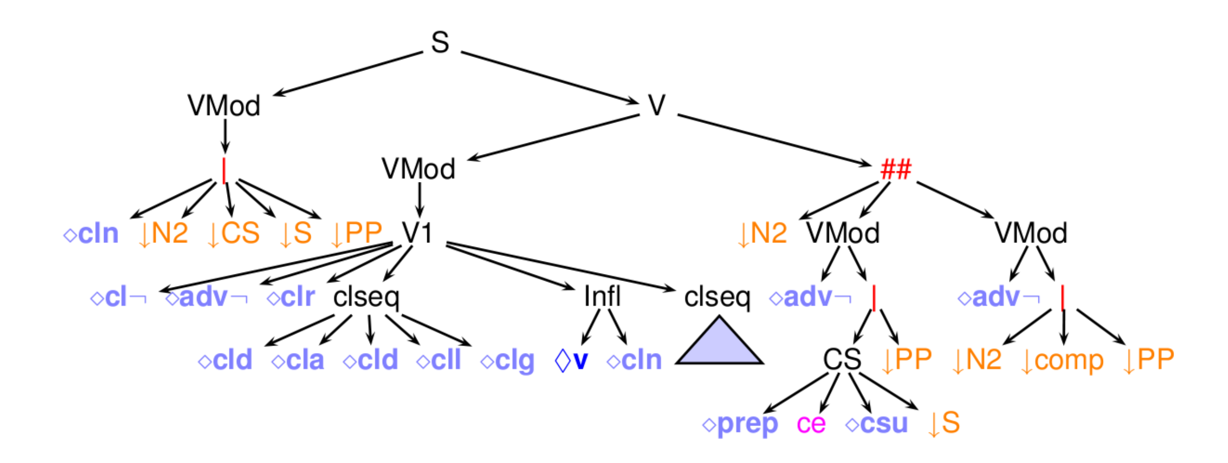


FIGURE 4.4: Arbre #111 (simplifié)

Un arbre, tel que celui de la Fig. 4.4, couvre la réalisation de nombreuses constructions syntaxiques. Pour cela, on attribue un hypertag à l'ancre des arbres ancrés (Thomasset et de La Clergerie, 2005). Cet hypertag décrit, par des structures de traits, un ensemble de mots du lexique qui réunissent certaines caractéristiques. Ainsi, un même arbre, comme celui de l'exemple donné, peut réunir la réalisation de plusieurs verbes qui partagent des comportements similaires. Cela permet de réduire considérablement le nombre d'arbres dans la grammaire. L'hypertag de l'ancre d'un arbre se spécialise ou se généralise, en fonction des informations décrites sur lui. Ces étiquettes sont créées pour chaque arbre

par DyALog (de La Clergerie, 2005a) à partir des restrictions décrites par les structures de traits présentes dans la décoration des nœuds et dans les équations des gardes de la méta-grammaire.

Par conséquent, le point d'ancrage entre les mots du texte, déjà étiquetés par FRMG *Lexer* avec des hypertags, et les arbres de la grammaire TAG, est réalisé par l'unification des structures de traits des étiquettes des mots avec celles associées aux arbres. Cette opération permet de sélectionner les arbres autorisés pour un mot particulier.

La Fig. 4.5 montre l'hypertag associé à l'arbre #111, alors que la Fig. 4.6 montre l'étiquette correspondant au verbe *promettre* présent dans le lexique Le*fff*.

Parmi tous les arbres présents dans la grammaire TAG du français, on sélectionne uniquement ceux dont les hypertags s'unifient correctement avec ceux assignés aux mots du texte d'entrée. Dans le cas présent, l'hypertag de la forme *promettre* s'unifie convenablement avec l'arbre #111 correspondant. Dans les deux étiquettes, on retrouve une construction verbale canonique, qui autorise un objet et un groupe prépositionnel introduit par la préposition *à*. Par conséquent, la réalisation de l'arbre proposé est permise par le mot *promettre*.

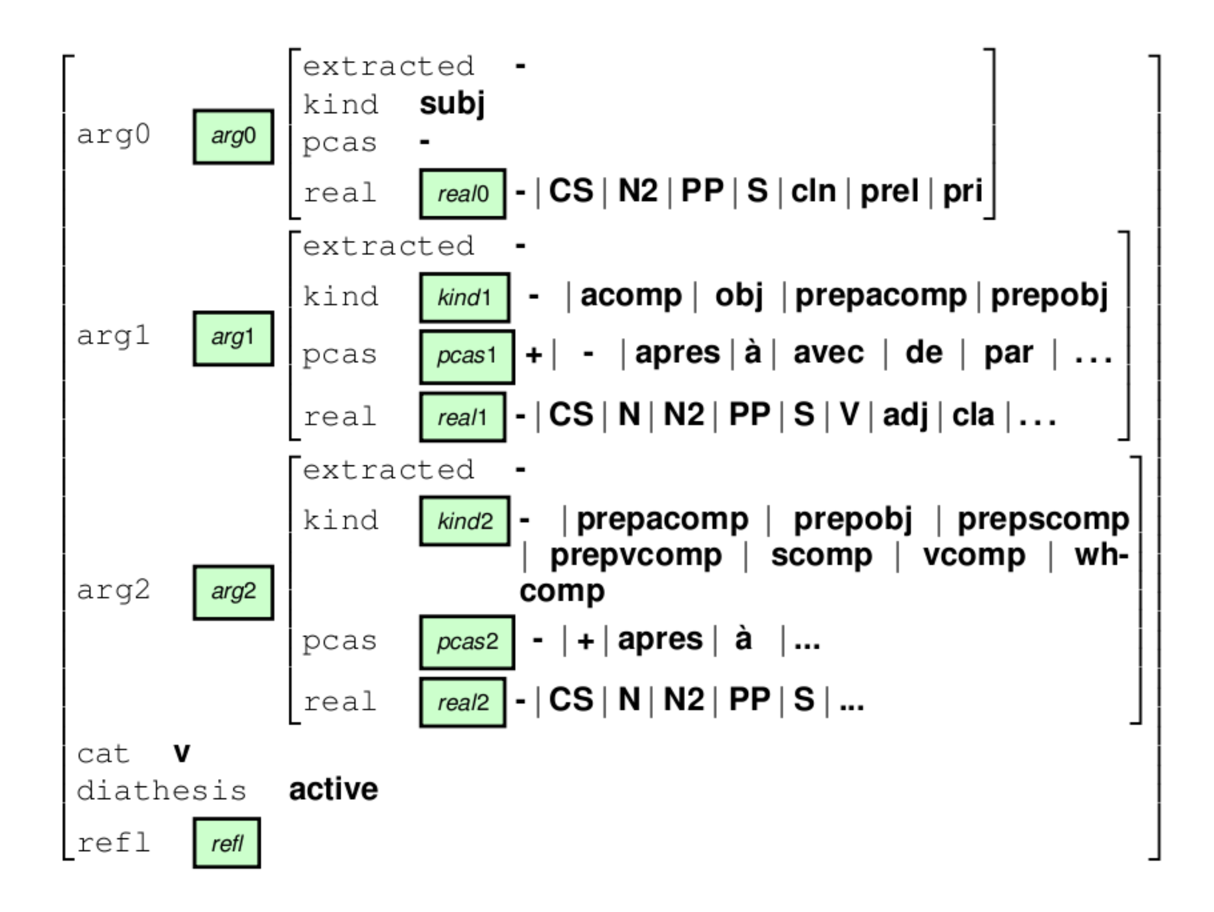


FIGURE 4.5: Hypertag de l'arbre #111

```
[arg0  [kind  subj | -      ]                ]
[      [pcas  -             ]                ]
[arg1  [kind  obj | scomp | -]               ]
[      [pcas  -              ]               ]
[arg2  [kind  prepobj | -   ]                ]
[      [pcas  à | -         ]                ]
[refl  -                                     ]
```

FIGURE 4.6: Hypertag du mot *promettre* dans le lexique Le*fff*

Les co-ancres lexicales sont également permises. Il s'agit de permettre l'ajout dans les arbres TAG lexicalisés, d'autres points d'ancrage avec le lexique à part l'ancre principale. Ce sont les co-ancres, qui, contrairement au point d'ancrage principal par les hypertags, se font directement sur le lexique [15].

À partir de la grammaire TAG factorisée pour le français, DyALog crée un analyseur syntaxique pour le français, reposant sur une stratégie d'analyse tabulaire descendante de la gauche vers la droite : FRMG *Parser*. Le processus d'analyse syntaxique implique les étapes suivantes :

1. **Ancrage lexical-syntaxique** : On sélectionne l'ensemble des arbres de la grammaire TAG dont les hypertags des ancres s'unifient correctement avec les mots du texte d'entrée. Au terme de cette étape, on dispose d'un ensemble d'arbres liés aux mots remplissant les fonctions syntaxiques principales de la phrase d'entrée [16]. Cependant, les mots avec des rôles secondaires, tels que les déterminants ou les prépositions, ne se lient avec aucune structure [17].
2. **Construction de la structure syntaxique** : À partir de cet ensemble d'arbres TAG, on essaye de construire la structure syntaxique qui couvre le texte d'entrée, par des opérations de substitution et d'adjonction. Idéalement, à la fin de ce processus, on obtient un arbre dérivé unique. En cas d'ambiguïté syntaxique, cette phase produira plusieurs arbres dérivés décrivant d'une manière différente la structure entière du texte d'entrée.
3. **Décoration de la structure syntaxique** : L'arbre dérivé résultant (ou les arbres dérivés) se trouve décoré seulement par les ancres des arbres élémentaires impliqués. Par conséquent, à ce stade, on insère les mots secondaires du texte d'entrée dans les nœuds feuilles de l'arbre final. La décoration s'établit selon l'unification,

15. Par exemple, l'arbre modélisant la comparaison *Jean est plus grand que Paul*, contient la co-ancre lexicale *que*, étant donné que ce mot se répète dans toutes les constructions de comparaison.

16. Il peut s'agir, par exemple, de substantifs comme noyau d'un syntagme nominal ou de verbes comme noyaux de la phrase.

17. Sauf si elles ont un rôle décisif dans une certaine construction, une préposition pouvant, par exemple, être l'ancre qui détermine la structure d'un syntagme prépositionnel. L'importance d'un mot ne dépend pas de sa catégorie lexicale, mais de la position qu'il occupe dans la structure analysée.

entre les restrictions incluses dans les nœuds feuilles et les hypertags des mots secondaires impliqués. Elle est réalisée par substitution.

4. **Construction de la sortie** : L'analyse syntaxique obtenue par FRMG *Parser*, est retournée sous la forme d'un arbre de dérivation, qui détaille les opérations effectuées pour construire la structure syntaxique de la phrase d'entrée. En cas d'ambiguïté syntaxique, le résultat est une forêt partagée de dérivations. Celle-ci regroupe les arbres de dérivation de chacune des structures syntaxiques acceptées par le texte analysé. Pour cela, une forêt détaille la structure commune des arbres de dérivation résultants et les variations structurelles de chacun d'entre eux.

En outre, DyALog, permet la construction d'analyseurs robustes pouvant fournir, lorsqu'une analyse complète n'est pas possible, l'ensemble des analyses partielles couvrant au mieux le texte d'entrée (de La Clergerie *et al.*, 2009). Ainsi, ce système offre toujours à l'utilisateur une analyse syntaxique, qu'elle soit complète ou partielle.

Notons qu'afin d'améliorer l'utilité de l'analyseur syntaxique, on peut lui imposer une limite de temps, au bout de laquelle les réponses trouvées sont émises même si les calculs ne sont pas finis.

4.2.3 Représentation de l'analyse : *Forest utils*

La ressource *Forest utils*, incluse dans la chaîne de traitement, permet de traiter la sortie de l'analyse, de sorte qu'elle soit utile et compréhensible pour les utilisateurs potentiels de l'application : un utilisateur humain ou une application de TAL de haut niveau.

Pour que l'analyse fournie par la chaîne en construction puisse être utilisée par des applications de TAL de haut niveau, *Forest utils* représente la forêt de dépendances au format XMLDep (Thomasset et de La Clergerie, 2005). Ce format utilise le standard XML pour décrire les dépendances syntaxiques résultant de l'analyse du texte d'entrée. Les principaux éléments présents dans le format XMLDep sont les suivants :

– **Nœud et Cluster** : Les clusters, ou groupes de nœuds, représentent les formes présentes dans la phrase analysée. Dans chaque cluster, se trouvent des nœuds étiquetés par un lemme possible de la forme associée au cluster correspondant. À son tour, chaque nœud est décoré, entre autres, avec la catégorie lexicale du lemme, le numéro d'identification de l'arbre TAG dont ce mot est ancre, et un ensemble de dérivations.
– **Arc** : Les arcs lient un nœud source à un nœud destination et sont décorés par une étiquette indiquant la dépendance syntaxique entre les deux nœuds.

Fondamentalement, on essaye de représenter au format XML un graphe de dépendances constitué de nœuds (lemmes), regroupés en clusters (formes), avec des arcs décrivant les dépendances syntaxiques entre les nœuds.

Pour que les dépendances syntaxiques représentées en XML soient compréhensibles pour un utilisateur humain, il est nécessaire de les transformer en un graphe de

dépendances (de La Clergerie *et al.*, 2009).

Le graphe de dépendances de la phrase *Jean promet une chanson à Agathe* est montré à la Fig. 4.7. Il s'agit d'une représentation graphique du format XMLDep, où les nœuds sont décrits à travers des ellipses jaunes. L'information présente dans chaque nœud est le lemme associé, sa catégorie lexicale et le numéro de l'arbre ancré qu'il représente. Entre parenthèses apparaît le nombre de dérivations en ce nœud. Les nœuds se trouvent inclus dans des clusters, dessinés sous forme de rectangles.

Les dépendances syntaxiques sont représentées par des arcs dirigés et étiquetés par la fonction syntaxique correspondante. Entre parenthèses on indique les dérivations du nœud qui correspondent à cette dépendance. Les arcs de couleur rouge représentent l'opération d'adjonction sur l'arbre indiqué dans le nœud d'origine de l'arc dirigé. Ainsi, dans la Fig. 4.7, l'arbre #25 du nœud *S* est inséré par adjonction dans l'arbre #198 du nœud *promettre*. Les arcs de couleur bleue représentent une opération de substitution sur l'arbre du nœud d'origine de l'arc dirigé. Ainsi, l'arbre #59 du nœud *chanson* est inséré par substitution dans l'arbre #198 du nœud *promettre*. Un autre type d'arc possible, qui n'apparaît pas ici, est celui de couleur violette, lequel fait référence aux co-ancres lexicales.

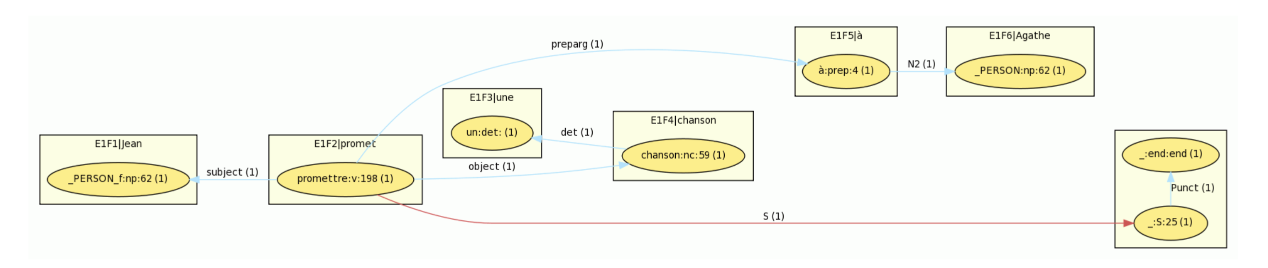


FIGURE 4.7: Graphe de dépendances pour *Jean promet une chanson à Agathe*

Un autre composant important de la chaîne de traitement est le désambiguïsateur, écrit en DyALog, qui permet de ne considérer qu'une analyse par phrase en éliminant l'ambiguïté. Il s'appuie sur un algorithme en programmation dynamique de recherche de la meilleure analyse en sommant les poids des arcs (et, dans une bien moindre mesure, des nœuds) participant à une analyse. Le poids d'un arc résulte de l'accumulation des poids donnés par des règles élémentaires exprimées sous forme de motifs prenant en compte l'arc courant (nœuds source et cible, type, label) et éventuellement les arcs frères, fils ou parents, voire des arcs en compétition. Les poids sont choisis de manière heuristique et ne résultent pas d'un processus d'apprentissage. Le désambiguïsateur comporte plus d'une centaine de règles élémentaires. Par exemple, citons l'existence de règles favorisant les arcs remplissant la valence d'un verbe (sujet, objet, etc.), la présence d'un sujet avant son verbe, l'inversion du sujet si certaines conditions sont remplies, etc. D'autres règles pénalisent les dépendances à longue distance, les transcatégorisations non nécessaires, certaines constructions improbables, etc.

Le coût théorique de l'algorithme est polynomial, conduisant en pratique à des temps de désambiguïsation très longs sur certaines phrases très ambiguës. En moyenne, les temps de désambiguïsation sont du même ordre de grandeur que les temps d'analyse

et donc non négligeables. Par ailleurs, la désambiguïsation est relativement instable, pouvant dramatiquement dépendre de variations sur les poids.

Enfin, dans le cadre des campagnes EASy et Passage, le désambiguïsateur est complété par un convertisseur, écrit en DyALog, vers les formats attendus par les organisateurs. Ces deux formats, très proches, proposent une annotation syntaxique sous forme de groupes non récursifs (GN, GA, GR, GP, NV, PV) et de relations (SUJ-V, AUX-V, COD-V, ATB-SO, CPL-V, MOD-V, MOD-N, MOD-A, MOD-R, MOD-P, COORD, APPOS, JUXT, COMP) entre formes et/ou groupes (cf. 2.1.2). Ces formats sont prévus pour des analyses plus surfaciques que celles rendues par FRMG, amenant à des pertes d'information et à des erreurs pendant la conversion.

Enfin, précisions que FRMG est un logiciel libre, tout comme le Le*fff*, accessible sous la GForge de l'INRIA [18]. Il est également possible de jouer avec la chaîne de traitement et de visualiser la grammaire FRMG sur http://alpage.inria.fr/frmgdemo.

18. http://gforge.inria.fr/projects/mgkit/

Deuxième partie

Modifications des tables

5

Amélioration des tables et création des tables des classes

Ce chapitre décrit avec plus de précision les tables du Lexique-Grammaire, l'intérêt de leur format et les améliorations que nous y avons apportées. Nous avons inclus tout au long de ce chapitre de nombreux exemples, permettant de se familiariser avec les notations des tables. Notons qu'un exemple précédé du signe * désigne une phrase non acceptable (ou éventuellement acceptable dans un emploi différent de celui qui est envisagé).

Tout d'abord, la section 5.1 contient un inventaire précis des tables, en indiquant leur provenance et en effectuant une sorte d'état des lieux, soulevant l'origine des problèmes de cohérence que les tables contiennent. Ensuite, la section 5.2 met en avant le format d'origine des tables, tout en expliquant leur classification. Puis, nous expliquons comment nous avons formalisé et homogénéisé les propriétés syntaxico-sémantiques dans la section 5.3, en listant les différentes solutions apportées. Dans la section 5.4, nous définissons ce qu'est une table des classes, en expliquant son rôle. Enfin, nous détaillons le codage qu'il reste encore à faire dans la section 5.5. Pour finir, la section 5.6 montre comment l'exploitation des tables du Lexique-Grammaire dans des systèmes de TAL est possible à partir de la version actuelle des tables.

5.1 Provenance des tables et état des lieux

Nous avons commencé par rassembler les informations sur les tables actuellement développées, afin d'en faire un inventaire aussi complet que possible (Tolone, 2009),

figurant dans la Tab. 5.1 [1]. Nous les avons classées par catégories (verbes distributionnels, noms prédicatifs, expressions figées [2], adverbes, adjectifs prédicatifs et adjectifs non prédicatifs). Pour chaque ensemble de tables nous indiquons :

- leur provenance (avec les références des publications concernées) ;
- le nom des tables (qui n'est pas une liste complète en soit mais qui permet de les identifier en respectant l'ordre alphabétique [3]) ;
- le nombre de tables concernées (avec entre parenthèses les éventuelles adjonctions récentes détaillées par la suite).

Les tables du Lexique-Grammaire n'ont pas exclusivement été conçues pour être exploitées dans des applications de TAL au début de leur élaboration, vers 1968. Toutefois, leur extension s'est poursuivie, s'est étendue à d'autres langues que le français, et elles restent actuellement utilisées. Cela a fait prendre conscience, à certains des auteurs, des potentialités d'exploitation incluses dans les tables du Lexique-Grammaire. C'est pourquoi, l'équipe d'informatique linguistique du LIGM de l'Université Paris-Est s'est récemment investi dans cette exploitation, notamment grâce à cette thèse et au projet *LGTag*, qui a impliqué Éric Laporte, Christian Leclère, Stavroula Voyatzi, Takuya Namakura, Matthieu Constant et moi-même. Durant ma thèse, j'ai joué un rôle de coordination entre ces contributeurs, en plus de ma contribution individuelle.

En effet, la plupart des tables constituent autant de fichiers Excel, qui ne forment pas véritablement un tout cohérent. Outre les entrées non encore codées (des signes $\sim$ remplacent alors les $+$ et les $-$), ces tables souffrent de diverses formes d'incohérence et d'incomplétude :

- certains intitulés de colonnes diffèrent d'une table à l'autre, bien qu'ils dénotent la même propriété linguistique ;
- certains intitulés de colonnes ne dénotent pas la même propriété d'une table à l'autre ;
- certains intitulés de colonnes laissent implicites des informations, pourtant nécessaires à leur exploitation automatique ;
- des informations importantes ne sont pas représentées, car elles sont considérées comme implicites pour une table donnée.

Tous ces problèmes proviennent du fait que les tables ont été élaborées durant plus de 40 ans par différents auteurs. Elles font souvent partie d'annexes de thèses, certaines, mais pas toutes, ont été reprises ensuite dans la publication de livres. D'autres figurent dans des revues (telles que Lingvisticæ Investigationes ou les Cahiers de Lexicologie) ou des rapports (Rapport de recherche ou Rapport technique du LADL). Chaque auteur apporte sa contribution dans le cadre de son travail de recherche, avec sa propre vision et ses propres notations, tout en respectant le même système de codage ($+$ et $-$), représentant l'acceptation ou non de propriétés syntaxico-sémantiques par les entrées

1. La Tab. 5.1 est reprise à la Tab. 9.1 afin d'y ajouter les différents formats auxquels les tables ont été converties.

2. Les expressions figées ne forment pas en soit une catégorie, mais réunissent un ensemble de catégories, surtout la table C0E, comme nous le verrons en 6.4.1.

3. La liste complète des tables avec le nombre d'entrées par table figure dans la section 6.6.

Références	Nom des tables	Nombre	Origine
(a) Verbes distributionnels (simples)			
Gross (1975)	1 à 18	17 (+1)	Livre
Boons *et al.* (1976a) (BGL)	32A à 32R3, 36R, 38PL, 38R et 39	15 (+1)	Rapport
Boons *et al.* (1976b) (BGL)	31H, 31R, 33, 34L0, 35L à 35R	8 (+3)	Livre
Guillet et Leclère (1992) (BGL)	36DT, 36SL, 37E à 38LR	16 (+3)	Livre
Leclère (1990), Borillo (1971) (BGL)	36S	1	Revue
Leclère (1990) (BGL)	35RR et 38RR	2	Revue
(b) Noms prédicatifs (simples et composés)			
Giry-Schneider (1978)	F1A à F91	21	Livre
Giry-Schneider (1987)	FN à FNPNN	10	Livre
Meunier (1981)	AN01 à AN06 et ANSY	7	Thèse
G. Gross (1989)	AA, AD, DR1 à ES, FR1 à IS2	15	Thèse
Giry-Schneider et Balibar-Mrabti (1993)	AN07 à AN10	4	Rapport
Giry-Schneider (2005b)	ANDN et ANSN	2	Revue
de Négroni-Peyre (1978)	PSY et SYM	2	Revue
Vivès (1983)	APE1 à APP3	9	Thèse
Labelle (1974)	ANA, ANM à ANS et ANSU	8	Thèse
Pivaut (1989)	FD1 à FD4	5	Thèse
(c) Expressions figées (principalement verbales et adjectivales)			
Boons *et al.* (1976b) (BGL)	31I	1	Livre
Gross (1982)	A1 à E0P1, E01, ECO et YA	38	Revue
Gross (1988)	EAPC, ENPC et EPA	3	Revue
Gross (1996)	EPAC à EPDETC	9	Revue
M. Gross	E1, E1HC, E1PN et EDN	4	Non publié
Giry-Schneider (1987)	FC à FCPNN	9	Livre
Danlos (1980)	Z à ZS	8	Thèse
(d) Adverbes (simples et (semi-)figés)			
Molinier et Levrier (2000)	ADVMF à ADVPS	16	Livre
Gross (1986a)	PAC à PVCO	16	Livre
(e) Adjectifs prédicatifs (simples)			
Picabia (1978)	ADJ1 à ADJ13	13	Livre
J. Giry-Schneider	ADJ01 à ADJSYM	38	Non publié
(f) Adjectifs non prédicatifs (simples)			
Laporte (2005)	ADJLOCTABLE	1	Revue
Giry-Schneider (2005a)	DEDJA et DADJI	2	Revue

TABLE 5.1: Inventaire des tables du Lexique-Grammaire

lexicales.

Il faut en effet souligner la cohérence partielle de l'ensemble, malgré le nombre d'auteurs et la durée de construction. Citons par exemple les notations présentées en section 3.1, qui sont issues de conventions assez bien respectées par les différents auteurs. C'est une prouesse étant donné les habitudes individualistes des linguistes. Il n'existe aucune autre réalisation atteignant à la fois le même niveau de cohérence, la même couverture lexicale et la même couverture grammaticale.

Des travaux de mise en cohérence et d'explicitation des propriétés inventoriées dans les tables du Lexique-Grammaire ont été mis en place au LIGM, en ayant comme objectif de les rendre exploitables dans un analyseur syntaxique. Les méthodes adoptées afin de résoudre les problèmes qui se sont posés lors de cette étape d'homogénéisation des tables, constituent l'objet de ce chapitre.

5.2 Format d'origine et classification

Dans cette section, nous expliquons en quoi le format sous forme de tables est plus lisible que d'autres, et pourquoi les intitulés de propriétés doivent être succincts et ne pas prendre la forme de structures de traits (5.2.1). Ensuite, nous précisons que chaque classe représente une sélection des entrées possédant un ensemble de *propriétés définitoires*, expression que nous définissons et illustrons à travers toutes ses représentations possibles (5.2.2). Lors de la délimitation des entrées lexicales, nous expliquons pourquoi les entrées sont séparées en cas de polysémie. De plus, nous présentons les tables qui présentent des doublons, correspondant à des descriptions croisées par différents auteurs (5.2.3). Enfin, nous listons les différentes versions existantes en détaillant le format Excel actuel, pour lequel il a fallu supprimer des cartouches horizontaux et numériser des tables de noms (5.2.4).

5.2.1 Format tabulaire pour une meilleure lisibilité

Les tables ont été conçues pour être lisibles (Laporte, 2010), le but étant d'avoir un format facilement manipulable par des linguistes n'étant pas informaticiens. En effet, les entrées lexicales (en lignes) sont faciles à identifier visuellement et à comparer. De même, les propriétés syntaxico-sémantiques matérialisées verticalement (en colonnes) sont identifiées par les valeurs qu'elles prennent. Ces valeurs sont représentées par des *étiquettes*, ou *intitulés*, répétés sur chaque écran, et donc très succincts (chaque intitulé occupe au plus une trentaine de caractères). Ce format tabulaire permet de croiser sur un même écran des dizaines d'entrées avec des dizaines de propriétés. Ainsi, lorsque le linguiste code une entrée, il a sous les yeux la description d'entrées comparables, pour peu que chaque table regroupe une classe d'entrées suffisamment homogène. Cette visualisation facilite le codage, qu'il soit seul ou en groupe, et ce format favorise également l'échange entre les linguistes.

Chaque propriété syntaxico-sémantique décrit partiellement une *construction*. L'intitulé N0 être Vpp, dans la table 31H de BGL (pour toutes les tables de verbes distributionnels citées, se référer à la Tab. 5.1(a)), représente une phrase à interprétation statique constituée du sujet N0 de la construction de base, du verbe *être* et du participe passé du verbe. Dans le cas de l'entrée *s'évanouir*, dont la construction de base est illustrée par *Luc s'évanouit*, la construction à sens statique en question est celle de *Luc est évanoui*.

Pour que les intitulés de propriétés soient mnémoniques, ils ont été construits à partir de symboles représentant des valeurs de traits : N pour substantif ou argument syntaxique, être pour le verbe *être*, pp pour participe passé. Mais pour qu'ils soient succincts, on n'y a généralement pas précisé les traits correspondants, respectivement ici : catégorie grammaticale, verbe support, temps/mode.

Avant d'exploiter les tables du Lexique-Grammaire dans un système de TAL, il est important de s'assurer qu'il peut être achevé et mis à jour, et donc qu'il existe sous un format que l'on peut aisément lire et éditer. Pour certains (Gardent *et al.*, 2005, 2006), le format des tables du Lexique-Grammaire n'est pas standard, car les constructions ne sont pas sous la forme de structures de traits, avec noms de traits et noms de valeurs, comme celles utilisées par les systèmes actuels. En adoptant de telles conventions, la propriété ci-dessus serait représentée par des formules telles que (Laporte, 2010) :

```
construction:[predicate:[part-of-speech="verb",
                         mode="participle",
                         tense="past"],
              support-verb:[part-of-speech="verb",
                            lemma-list:[value="être"]],
              arguments:(constituent:[position="0",
                                      distribution:[component:[category="NP"]]
                                                                          ])]
```

Ou, par des formules équivalentes en XML, encore moins concises, même si elles sont adaptées aux programmes informatiques :

```
<construction>
  <predicate>
    <part-of-speech value="verb" />
    <mode value="participle" />
    <tense value="past" />
  </predicate>
  <support-verb>
    <part-of-speech value="verb" />
    <lemma-list value="être" />
  </support-verb>
  <arguments>
    <constituent position="0">
      <distribution>
        <category value="NP" />
```

```
      </distribution>
    </constituent>
  </arguments>
</construction>
```

Les structures de traits sont un standard destiné à d'autres usages, incompatibles avec les exigences de compacité et de lisibilité de l'édition manuelle des tables. Les projets ComLex Syntax (Grishman *et al.*, 1994) et FrameNet (Baker *et al.*, 1998) n'ont également pas adopté un format de structures de traits pour l'édition et la mise à jour des lexiques. Or, ce sont les projets producteurs de lexiques à grande couverture pour le TAL, tels que la construction des tables du Lexique-Grammaire, qui ont vocation à nourrir la construction des standards et des normes. C'est le sens de la réflexion sur le format des tables du Lexique-Grammaire effectuée par le projet Genelex (Alcouffe *et al.*, 1993), qui fut une des sources du projet de normalisation Eagles [4]. Le projet LexSynt [5] a également donné l'occasion de tenir compte des tables du Lexique-Grammaire lors de l'élaboration de la norme LMF (Francopoulo *et al.*, 2006).

5.2.2 Découpage en classes

Nous définissons les *propriétés définitoires*, en les illustrant à travers toutes leurs représentations possibles : constructions de base, propriétés distributionnelles, autres constructions, etc. Puis, nous introduisons ce que l'on appelle les *formules définitoires*, qui permettent de justifier la classification, avant d'expliquer les motivations et les problèmes de ce découpage.

Propriétés définitoires

Les tables du Lexique-Grammaire répartissent les entrées lexicales dans des classes. Chaque classe regroupe un certain nombre d'entrées jugées similaires, car elles acceptent des propriétés syntaxico-sémantiques communes, que l'on appelle les *propriétés définitoires* [6]. Les propriétés définitoires de ces classes relèvent généralement du cadre de sous-catégorisation. Ainsi, les critères les plus communément utilisés dans les propriétés définitoires sont le nombre de compléments, la nature prépositionnelle ou non des compléments (pour les compléments prépositionnels, sont distingués ceux qui sont introduits par les prépositions *à*, *de*, *avec*, Loc, etc.), la catégorie grammaticale du sujet et des compléments (sont distinguées les réalisations sous forme de complétive, notée Qu P, d'infinitive, notée V-inf W, et de syntagme nominal, notée N suivi d'un trait sémantique, comme par exemple, Nhum ou N-hum).

Par exemple, la table 9 de M. Gross a parmi ses propriétés définitoires : N0 V N1 à N2 (dans cette construction, N0 représente le sujet, V le verbe, N1 le premier argument, N2

4. http://www.ilc.cnr.it/EAGLES/home.html
5. http://lexsynt.inria.fr/index.php
6. Notons que le terme de propriété définitoire a été préféré à celui de *propriété définitionnelle*, mais ils sont tous les deux employés dans la littérature de manière équivalente.

le deuxième), où le complément essentiel direct N1 peut être occupé par une complétive : cette table regroupe des verbes comme *dire*, *dissimuler* et *ordonner*, dont le cadre de sous-catégorisation peut se caractériser par une complétive objet et un complément nominal introduit par la préposition *à* (*Luc a (dit+dissimulé+ordonné) à Marie que Zoé chante*). On peut remarquer que à N2 apparaît avant N1, en accord avec le fait que les propriétés définitoires n'imposent pas d'ordre sur les compléments. Le fait que l'argument N1 puisse être une complétive est codé dans la table pour différencier les complétives à l'indicatif (N1 =: Qu Pind), au subjonctif (N1 =: Qu Psubj) et celles qui sont interrogatives (N1 =: si P ou si P), et pour permettre également de reconnaître d'autres catégories grammaticales (voir 6.1.3).

La table 5 de M. Gross a parmi ses propriétés définitoires : N0 V Prép N1, mais aussi N0 =: Qu P, ce qui signifie que le sujet peut être occupé par une complétive. Ici c'est la préposition qui est codée dans la table, car elle varie en fonction des entrées (*Qu'Ida allait partir cheminait* **dans** *sa tête* / *Que Max s'est enfui circule* **sur** *son compte*). Le cadre de sous-catégorisation est donc défini par une complétive sujet et un complément nominal introduit par la préposition spécifiée dans la table.

La table 37M1 de BGL, a parmi ses propriétés définitoires : N0 V N1 Prép N2 et Prép2 =: de[7]. Contrairement à la table 5, toutes les entrées acceptent la préposition *de* pour l'argument N2 (*Max abrutit ses élèves* **de** *travail* / *On a doté l'hôpital* **de** *scanners*), mais aussi d'autres prépositions qui sont codées dans la table (*Max abrutit ses élèves* **avec** *du travail* / *On a doté l'hôpital* **en** *scanners*). La table 9 n'accepte que la préposition *à* pour l'argument N1, c'est pourquoi la propriété définitoire N0 V N1 à N2 suffit ici pour préciser à la fois la préposition et le nombre d'arguments. Le cadre de sous-catégorisation pour la table 37M1 est un complément nominal objet et un complément nominal introduit par la préposition *de*, mais aussi *avec* ou *en*, etc., selon les entrées.

Les propriétés définitoires sont constituées d'au moins une construction, dite *construction de base*. Ainsi, N0 V N1 à N2 est la construction de base de la table 9, alors que N0 V Prép N1 est celle de la table 5 et enfin, N0 V N1 Prép N2 celle de la table 37M1. La propriété N0 =: Qu P définitoire de la table 5 est une *propriété distributionnelle*, qui spécifie la catégorie grammaticale de l'argument N0, déjà défini dans une construction, souvent la construction de base. La propriété Prép2 =: de est également une propriété distributionnelle, définitoire de la table 37M1, qui spécifie la valeur de la préposition[8].

Une propriété définitoire peut aussi indiquer qu'un élément de la table entre dans deux constructions, qui sont généralement reliées par un lien de paraphrase. Cette deuxième construction est appelée *propriété transformationnelle*, car elle est déductible de la première par une redistribution, la première étant la construction de base. Ainsi la table 35S regroupe les verbes intransitifs *symétriques* qui se caractérisent par la construc-

7. Remarquons que Prép2 fait référence à la préposition du deuxième complément, même si elle n'est pas numérotée dans la construction.

8. Remarquons que la construction de base de la table 9 pourrait être également N0 V N1 Prép N2, avec Prép2 =: à définitoire, mais l'on a préféré intégrer la préposition dans la construction de base lorsqu'une seule était possible.

tion de base N0 V Prép N1 et par la deuxième construction définitoire N0 et N1 V (*Luc flirte avec Zoé / Luc et Zoé flirtent (ensemble)*).

Enfin, de nombreuses propriétés définitoires incluent des traits sémantiques élémentaires. Par exemple, des informations sur les classes des noms têtes des syntagmes nominaux (humain, concret, pluriel, etc.) : ainsi, la table 31H admet la propriété distributionnelle N0 =: Nhum indiquant que le sujet N0 de la construction de base doit être occupé par un groupe nominal dénotant une entité humaine (*Luc s'évanouit*).

Ou encore, des informations sur la sémantique des procès : ainsi les verbes entrant dans la construction N0 V N1 Prép N2, avec Prép =: de et dont le N1 dénote soit un lieu par rapport auquel est située l'entité à laquelle réfère N2, soit le bénéficiaire ou le détrimentaire du procès, ont été divisés en deux ensembles : la table 37E de M. Gross regroupe les procès d'enlèvement (*Luc a débarrassé le grenier de ses caisses*, exprimant l'acte d'enlever les caisses), tandis que les tables 37M1 à 37M6 de M. Gross regroupent les procès d'ajout (*Luc a muni la porte d'un verrou*, exprimant l'acte de mettre un verrou)[9]. Les tables 37M1 à 37M6 se distinguent par des propriétés très diverses (morphologiques, sémantiques ou autres) qui ne relèvent pas du cadre de sous-catégorisation et qui auraient pu figurer en colonne dans une unique table 37M. La raison de ce découpage est principalement numérique : la table 37M aurait regroupé 890 verbes, et il a été considéré que la consultation manuelle d'une matrice de 890 lignes était laborieuse, d'où sa division en six sous-tables (voir 6.2.2).

Dans tous les cas cités précédemment, les propriétés définitoires correspondent à des propriétés qui sont vraies pour toutes les entrées d'une table, mais l'inverse est possible également, à savoir des propriétés interdites pour l'ensemble des entrées d'une table.

Reprenons, par exemple, la table 9 qui a comme construction de base N0 V N1 à N2, avec une complétive en N1. Il faut exclure la possibilité d'une complétive en N2, sinon cela inclurait la table 16 par exemple, qui a comme construction de base N0 V Prép N1 Prép N2, avec une complétive en N1 et en N2.

De même, la délimitation de la table 37M1 (N0 V N1 Prép N2, avec Prép =: de) inclut la négation de propriétés de complétives, sinon par exemple une partie de la table 10, qui a comme construction de base N0 V N1 Prép N2, avec une complétive en N1 (*Le maire requiert du préfet que la police intervienne*), se trouve incluse dans la table 37M1.

C'est le cas également de la table 32NM qui n'accepte pas la *redistribution* passive (*Cette valise pèse 10 kilos / *10 kilos sont pesés par cette valise*). Par exemple, on peut interdire les propriétés [passif par] et [passif de], utilisées selon que le complément d'agent est introduit par la préposition *par* ou *de*, et qui indiquent que tous les autres objets sont conservés au passif. Ces deux propriétés font partie des propriétés transformationnelles, puisque ce sont des redistributions à partir de construction de base.

Enfin, nous pouvons voir le cas de la table 32H, qui a la propriété distributionnelle N1 =: Nhum obligatoire, ce qui signifie que N1 =: Nhum est toujours vraie alors que N1 =: N-hum n'est jamais acceptée.

9. Ceci est représenté par les deux propriétés définitoires N0 V N1 de N2 = N0 enlever N2 de N1 et N0 V N1 de N2 = N0 mettre N2 Loc N1, comme nous le détaillerons en 6.2.2.

Formules définitoires

Si l'on reprend à nouveau la table 9, la possibilité d'avoir une complétive en N1 est codée dans la table par les trois colonnes N1 =: Qu Pind, N1 =: Qu Psubj et N1 =: si P ou si P. La disjonction de ces trois propriétés fait donc partie de la définition de la table. Cela signifie qu'il faut prendre en compte cet ensemble de propriétés séparées par des ou logiques, c'est-à-dire dont au moins une des propriétés parmi l'ensemble est vraie. Cela correspond pour la table 9, si l'on écrit formellement cet ensemble, à la formule booléenne suivante : (N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 =: si P ou si P).

De même, si l'on définit la possibilité d'avoir une complétive en N2 par les deux propriétés N2 =: Qu Pind et N2 =: Qu Psubj, l'exclusion de cette possibilité est représentée par les deux négations de propriétés non (N2 =: Qu Pind) et non (N2 =: Qu Psubj). La conjonction de ces deux négations de propriétés fait également partie de la définition de la table et peut être définie comme suit : non (N2 =: Qu Pind) et non (N2 =: Qu Psubj).

Nous pouvons ainsi définir formellement la délimitation d'une table par une *formule définitoire*, composée d'un ensemble de disjonctions, conjonctions et négations de propriétés, autrement dit, un ensemble de propriétés séparées par des ou logiques, des et logiques et des négations non. Elle y inclut aussi bien les propriétés définitoires vraies que fausses (introduites par la négation non) pour l'ensemble d'une table, ainsi que les disjonctions de propriétés codées dans la table (ensemble de propriétés séparées par des ou logiques). À titre d'exemple, reprenons l'ensemble des propriétés que l'on vient de citer pour la table 9 :

- sa construction de base est N0 V N1 à N2 ;
- l'exclusion de la possibilité d'avoir d'une complétive en N2 s'écrit :
  non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) ;
- la possibilité d'avoir une complétive en N1 s'écrit :
  (N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 =: si P ou si P)

On peut écrire formellement la définition de la table 9 (incomplète ici) par la conjonction de ces trois ensembles, ce qui donne la formule définitoire suivante :
(N0 V N1 à N2)
et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj)
et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 =: si P ou si P)).

Ce sont ces formules définitoires qui permettent de délimiter les différentes classes et d'avoir donc des classes disjointes. Elles ont été formalisées pour toutes les tables de verbes distributionnels [10], ce qui a permis de classer les tables selon un ordre de priorité à appliquer afin de savoir dans quelle table est incluse (ou doit être ajoutée) une entrée. L'ordre de priorité dans ce classement des verbes accompagné des formules définitoires de chaque table est inclus dans la version 3.3 et dans l'annexe F. Afin de pouvoir appliquer plus facilement cet ordre de priorité, nous avons créé un arbre de classement des verbes distributionnels, servant à déterminer à quelle classe appartient une entrée verbale donnée. Il est conçu pour un utilisateur qui connaît les propriétés de l'entrée et qui applique successivement les critères indiqués dans l'arbre. Il est inclus dans la version 3.3 et dans l'annexe G, accompagné d'une documentation expliquant les conven-

10. Nous avons réalisé ce travail en collaboration avec Eric Laporte et Christian Leclère en 2010-2011.

tions de lecture. Dans toute la suite, nous mettrons l'accent sur les propriétés définitoires vraies pour toutes les entrées d'une table, sans donner les formules définitoires.

Motivations et problèmes du découpage

Ceci nous amène à remarquer que ce découpage en classes, autrement dit le regroupement de certaines entrées, est en partie arbitraire. En effet, il est possible de prendre en compte des propriétés plus ou moins précises, qui englobent plus ou moins de verbes, et d'obtenir alors un nombre de classes différent. Par exemple, la construction N0hum V à N1 (qui revient à accepter la construction N0 V à N1 et la propriété distributionnelle N0 =: Nhum en même temps) concerne moins d'entrées que la construction N0 V à N1 sans restriction sur le sujet. De plus, d'autres propriétés auraient pu être prises en compte, ce qui aurait amené à un découpage totalement différent.

Mais ce qui importe, c'est le travail qui a été réalisé à partir de ce découpage. Nous aboutissons, au total, à 76 400 entrées prédicatives, qu'elles soient verbales, nominales, adjectivales ou adverbiales, mais également simples ou figées [11], classées selon les critères choisis. Cela constitue une très grande base de données lexicales, comparée aux autres lexiques, présentés en 2.2.2, qui ont en moyenne moins de 10 000 entrées prédicatives.

Cependant, les propriétés définitoires vraies pour toutes les entrées d'une table ne figurent pas dans les tables : ainsi, la construction de base de la table 9 est N0 V N1 à N2, mais la table n'a pas de colonne intitulée N0 V N1 à N2, car c'est une information implicite qui est uniquement décrite dans la littérature. Or, cette propriété sert de référence pour la représentation des autres constructions, comme la propriété transformationnelle N0 V à N2 (effacement du N1 : *Luc téléphone à tout le monde*), et pour les propriétés distributionnelles, comme N0 =: N-hum (sujet de type non humain : *Le chemisier blouse*).

Citons R. Vivès, qui résume bien la préoccupation qui à cette époque était liée à la taille des données (Vivès, 1983) : «Dans les tables que nous avons établies, nous avons cherché à fournir les indications les plus significatives en essayant de limiter le nombre des colonnes. Lorsqu'une propriété est générale ou quasi-générale pour l'ensemble de la table, nous ne l'avons pas représentée, nous contentant de la signaler dans les commentaires consacrés à chaque table. Lorsqu'une propriété n'est vérifiée que par un nombre réduit d'éléments dans une table, nous avons adopté la même solution : cette propriété fait l'objet d'un commentaire.»

Nous nous sommes donc référée aux commentaires de chaque table dans la littérature, afin d'ajouter toutes les propriétés définitoires vraies pour toutes les entrées d'une table (voir section 5.4).

5.2.3 Délimitation des entrées lexicales

Comme pour tout lexique au sens linguistique du terme, les objets de base des tables du Lexique-Grammaire sont les entrées lexicales. En cas de polysémie, les entrées sont

11. Sans compter les entrées adjectivales simples.

séparées : les entrées de *foncer* dans *Luc fonce au port* (reconnu par la construction de base N0 V Loc N1 de la table 2 de M. Gross) et dans *Le pigment fonce les couleurs* (décrit dans la table 37M4 avec la propriété N0 V N1 codée +) sont distinguées l'une de l'autre, de la même façon que *foncer* l'est de *fonder* dans *Luc fonde une agence* (reconnu par la construction de base N0 V N1 de la table 32A de BGL). Plusieurs constructions peuvent relever d'une même entrée. Ainsi, les tables du Lexique-Grammaire n'ont pas une entrée distincte pour *Les couleurs foncent* : il rattache cette construction à la même entrée que *Le pigment fonce les couleurs*, à travers la propriété N1 V W dans la même table 37M4.

Certaines classes font exception à ce principe. Ainsi, *Paul barbouille de la peinture sur le mur* est décrit dans la classe 38LD de BGL avec la construction de base N0 V N1 Loc N2 destination, où Loc désigne une préposition locative et l'objet locatif est interprété comme lieu de destination. La construction croisée (*Paul barbouille le mur de peinture*) est spécifiée dans cette entrée sous l'intitulé N0 V N2 (E+de N1), mais elle est également décrite indépendamment, de façon plus détaillée, dans la classe 37M6 (qui a pour construction de base N0 V N1 Prép N2 avec Prép2 =: de), avec une nouvelle numérotation. Il est prévu de rendre ces tables homogènes avec les autres sur ce point (Laporte, 2010).

De plus, le choix des propriétés définitoires a été fait individuellement par chaque auteur des tables du Lexique-Grammaire afin de correspondre à la thématique étudiée, sans forcément prendre en compte les autres travaux déjà réalisés, ou au contraire, en les incluant dans leur travail pour pousser plus loin leur description. Cela a conduit à ce que plusieurs entrées de même sens se retrouvent en doublons dans différentes classes, même si cela représente un cas minoritaire. Par exemple, pour les noms (cf. Tab. 5.1(b)), les tables FR1 à FR3 de G. Gross contiennent des doublons avec les tables de J. Giry-Schneider, ainsi que la table AA de G. Gross avec les tables d'A. Meunier (cela sera vu en détail en 6.3.3). Pour les expressions figées (cf. Tab. 5.1(c)), les entrées des tables de L. Danlos ont été pour la plupart reprises dans les tables de M. Gross, mais cela reste à vérifier. En ce qui concerne les verbes distributionnels (cf. Tab. 5.1(a)), certaines tables de BGL reprennent parfois des entrées déjà présentes dans les tables de M. Gross.

Ces entrées identiques sont donc considérées comme ayant deux sens distincts, puisqu'un même mot a autant d'entrées que de sens différents, alors que cela n'est pas le cas ici. Il faudrait donc fusionner les entrées redondantes, pour ne les faire figurer que dans une seule classe.

5.2.4 Format des tables

Le format Excel est celui utilisé pour l'édition des tables car il assure une stabilité aux tables dans le temps, puisqu'il permet de représenter les tables indépendamment du logiciel utilisé [12]. Dans un premier temps, nous avons cherché à obtenir toutes les tables au format électronique, puisque certaines tables étaient seulement disponibles dans la

12. Microsoft Excel ou des logiciels libres équivalents, tels que OpenOffice.org Calc, qui est téléchargeable sur le site http://www.openoffice.org/.

littérature (dans l'annexe d'ouvrages ou de thèses), c'est-à-dire en version 1. En effet, il existe plusieurs versions des tables que nous détaillons. Puis, nous expliquons pourquoi nous avons supprimé les cartouches horizontaux lors du passage au format Excel. Enfin, nous nous penchons sur le cas des tables que nous avons dû numériser et modifier, pour les rendre conformes aux autres tables.

Versions des tables

La version 1 est celle d'origine, c'est-à-dire celle qui figure sous format papier dans la littérature, mais qui existait également à l'époque sous format électronique, dans des cartes perforées (deux par table, l'une contenant les intitulés des propriétés et l'autre les entrées avec leur codage), en ce qui concerne les tables les plus anciennes. Au fil du temps, les tables ont subi divers changements de support électronique (cartes perforées, bandes magnétiques, disquettes) et de format (au fur et à mesure que les outils pour les éditer ont évolué) [13]. Certaines ont été perdues par manque d'intérêt de la part des auteurs, des institutions, ou de la communauté scientifique.

La version 2 représente 60% des tables informatisées qui ont été mises en ligne en 2002 par Nathalie Bely sur le site http://infolingu.univ-mlv.fr/ (Données Linguistiques > Lexique-Grammaire > Visualisation). Un système a été mis en place, permettant d'effectuer une recherche par verbe et par table, avec la possibilité d'afficher les exemples des verbes sélectionnés dans les tables sélectionnées. De plus, un téléchargement est possible en XML, et une documentation est associée à chaque table. Cette documentation indique les propriétés définitoires et donne des exemples pour une entrée lexicale afin d'illustrer les différentes propriétés.

Les récentes modifications ont donné jour à la version 3 en septembre 2008, également téléchargeable sur ce même site (Données Linguistiques > Lexique-Grammaire > Téléchargement), qui est régulièrement actualisée. La version 3.3 (31 mars 2011) contient la totalité des tables dans toutes les catégories, sous tous les formats disponibles (cf. chapitre 9). La version 3.3 contient de plus une documentation exhaustive sur toutes les propriétés syntaxico-sémantiques des verbes, ainsi que la définition formelle de chaque table et l'ordre de priorité dans le classement des verbes.

Suppression des cartouches horizontaux

Dans la version 1, des *cartouches* horizontaux matérialisent une classification des propriétés, autrement dit, certaines propriétés sont regroupées en familles, voire mises en dépendance les unes par rapport aux autres, comme illustré à la Fig. 5.1 (Boons *et al.*,

13. De même qu'avec les cartes perforées, deux fichiers au format texte permettaient de représenter une table, grâce au programme d'édition d'A. Guillet (nommé EDIX et enregistré sur deux disquettes MS-Dos) qui présentait tout dans le bon ordre. Chaque intitulé de propriété était sur une ligne dans le premier fichier et le programme les écrivait de telle sorte qu'on pouvait lire les intitulés en colonne. Le deuxième fichier contenait en ligne le numéro de la ligne, l'entrée, une barre oblique et une série de + ou − dans l'ordre par rapport aux intitulés énumérés dans le premier fichier. Voici par exemple, une ligne de la table 31H : 7 s'agiter/− − + − − + − + + + +).

1976b)[14]. On voit, en effet, que les six premières colonnes sont des propriétés distributionnelles de l'argument N0, comme l'indique le cartouche horizontal intitulé N0. De plus, l'intitulé N0 =: Npc (où Npc dénote le corps ou une partie du corps d'une personne) dépend de l'intitulé N0 =: Nhum, ce qui est représenté par le deuxième cartouche vertical inclus dans le premier.

Cette structuration n'est pas exploitable informatiquement de façon simple, car les intitulés de colonnes deviennent des objets complexes constitués de plusieurs étiquettes. De plus, même si le format Excel permet de garder les cartouches horizontaux grâce à des fusions de colonnes, ceci n'est pas conforme avec la contrainte d'avoir tous les intitulés sur la première ligne permettant d'utiliser les tables avec le logiciel Unitex (Paumier, 2003). Cette contrainte est d'autant plus valable aujourd'hui avec l'utilisation de l'outil *LGExtract*[15], qui permet de générer un lexique syntaxique pour le TAL, le lexique *LGLex*, à partir des tables du Lexique-Grammaire. Les cartouches horizontaux de la version 1 ont donc été supprimés, bien qu'ils aient contribué à la lisibilité tout en apportant des informations[16]. Lors de la suppression de ces cartouches et de leurs dépendances[17], les informations qu'ils contenaient ont été incorporées aux intitulés.

N_0											
N_0 = N hum	N_0 = N pc	N_0 = N -hum	N_0 = N nr	N_0 = V Ω	N_0 = V-n		N_0 est V-ant	N_0 est V pp	$N^0{}_{pc}$ lui V	N_0 V de $N^0{}_{pc}$	il V N_0 Ω
-	+	+	-	-	-	croupir	+	+	-	-	+
-	-	+	-	-	-	croustiller	+	-	+	-	+
-	-	+	-	-	-	cuver	-	+	-	-	+

FIGURE 5.1: Extrait de la table 31R des verbes distributionnels (version 1)

Un exemple est la table 36DT (version 1) (Guillet et Leclère, 1992), qui contenait l'intitulé Ppv =: lui dépendant de la colonne N2 =: N-hum. Il a été renommé Prép N2-hum = Ppv =: lui, la construction de base étant N0 V N1 Prép N2 (voir 6.2.2).

Un autre exemple est la table 34L0 (version 1) (Boons *et al.*, 1976b), où la numérotation des arguments de la moitié des constructions est indépendante de celle de la construction de base, et où deux cartouches horizontaux intitulés Standard et Croisée

14. La section 9.3 montre le même extrait de la table 31R sous tous les formats disponibles actuellement.

15. *LGExtract* est relié à une table globale dans laquelle nous avons ajouté les propriétés manquantes et un unique script d'extraction incluant toutes les opérations liées à chaque propriété devant être effectuées pour toutes les tables (voir chapitre 7).

16. Ce travail a été réalisé par Eric Laporte en 2003-2004 (Laporte, 2010). Nous avons reproduit le même travail en 2009 pour les nouvelles tables de noms numérisées (voir plus loin), ainsi que les nouvelles tables d'expressions figées numérisées (voir 6.4.4).

17. Les dépendances ne pouvant figurer dans le format Excel.

lèvent l'ambiguïté informellement. Ils correspondent respectivement à la construction de base N0 V Loc N1 (ou construction standard) et à la propriété transformationnelle N1 V de N0 (ou construction croisée), mais les propriétés associées à chaque cartouche sont toutes numérotées à partir de 0. On a donc un ensemble de constructions presque identiques sous chacun des 2 cartouches : par exemple, la propriété N0 est V-ant Prép N1 est codée deux fois, elle correspond à *La lumière est clignotante dans le salon* pour la construction standard, et à *La rue est clignotante de mille lumières* pour la construction croisée. Lors de l'élimination des cartouches horizontaux, les propriétés dépendantes du cartouche intitulé Croisée ont été renommées, en modifiant la numérotation de l'argument N0 (remplacé par N1) : par exemple, la propriété précédente a été renommée en N0 être V-ant Loc N1 pour la construction standard, et N1 être V-ant Prép N0 pour la construction croisée.

Ces conventions semblent avoir compliqué la compréhension des propriétés de certains utilisateurs. Ainsi, dans Gardent *et al.* (2005), on se demande si les indices (d'un argument dans une construction ou dans une propriété distributionnelle) font référence à la position du constituant dans la construction de base ou dans une autre.

Prenons le cas de la table 32PL de BGL pour expliquer ce problème : cette table a pour construction de base N0 V N1, mais dès la deuxième colonne codée dans la table, un argument numéroté N2 apparaît, par exemple dans l'intitulé N2 V N1. On ne peut pas relier cette redistribution à la construction de base, mais on peut la relier à une autre construction codée dans la table quelques colonnes après, intitulée N0 V N1 en N2. Il faut donc considérer la table dans son ensemble pour donner un sens à chaque constituant. Tous les intitulés étant à présent homogènes, les numéros d'ordre des constituants figurant dans les propriétés soit font référence à ceux figurant dans la construction de base, soit ce sont des arguments supplémentaires. Dans ce cas, ils font référence à une autre construction les contenant qui est plus longue que la construction de base. Informatiquement, cela ne pose pas de problème si l'on essaye toujours de se référer à la construction la plus longue. Ce point sera étudié plus en détail au chapitre 8.

Tables numérisées

Certaines tables de noms n'étaient disponibles qu'en version 1, nous les avons scannées et leur avons appliqué un outil de reconnaissance optique de caractères (OCR) [18], ce qui a ensuite nécessité un travail de correction manuelle.

Tout d'abord, une grande partie des entrées étaient en majuscules (par exemple, les tables ANA à ANS), il a donc fallu tout réécrire en minuscules, en ajoutant tous les accents car ils n'étaient pas présents. De plus, quelques fautes d'orthographe étaient déjà présentes dans les tables d'origine (comme par exemple pour la table SYM, les entrées *INTERVENTIR*, *JUMULER* et *RACCOM*, qui ont été corrigées en *intervertir*, *jumeler* et *raccommoder*). D'autres fautes sont apparues lors de la numérisation, car certains caractères étaient moins bien imprimés que d'autres, notamment des fautes d'accents

18. Nous avons réalisé ce travail en collaboration avec Jean-Pierre Heymann en 2009.

(comme *complicite* et *mëpris*, corrigés en *complicité* et *mépris* dans la table APE21) et quelques lettres mal reconnues (*accrOissement*, *équiiibre* et *rietus*, corrigés en *accroissement*, *équilibre* et *rictus* dans la table APP1) [19].

Ensuite, les cartouches horizontaux ont été éliminés lors de la correction manuelle. On a donc minutieusement renommé toutes les colonnes concernées pour rendre les intitulés conformes aux conventions des autres tables, ou identiques à ceux existants déjà lorsque leur définition correspondait, afin de former un tout cohérent. Par exemple, pour les tables APE1 à APE3, les 2 cartouches intitulés AVOIR et PERDRE englobaient chacun un ensemble de propriétés définissant la nature du déterminant. Prenons par exemple la propriété distributionnelle Det =: un, qui était présente à l'identique sous les 2 cartouches : elle a été remplacée par Det =: un, Vsup =: avoir et Det =: un, Vsup =: perdre pour spécifier avec quel verbe support se combine chaque déterminant codé.

Cela a permis d'avoir 16 tables de noms supplémentaires : les tables de D. de Négroni, R. Vivès et J. Labelle (cf. Tab. 5.1(b)). Leur description est faite en 6.3.4. La même chose a été faite pour certaines tables d'expressions figées : Z5D, Z5P, Z1, ZS et ZP de L. Danlos (cf. Tab. 5.1(c)). Nous étudierons cela en détail en 6.4.4.

5.3 Formalisation et homogénéisation des propriétés syntaxico-sémantiques

Après avoir défini les entrées et les propriétés définitoires, nous illustrons les propriétés syntaxico-sémantiques qui sont codées pour chaque entrée, en listant les différentes possibilités de représentation (5.3.1). Puis, nous montrons comment nous avons rendu toutes les propriétés homogènes et plus précises (5.3.2). Enfin, nous traitons de la création de la documentation complète de toutes les propriétés des verbes, ce qui a permis de les définir clairement (5.3.3).

5.3.1 Représentation des propriétés

Les tables se présentent sous forme de matrices de + et de − dans lesquelles les colonnes indiquent les propriétés syntaxico-sémantiques qui varient d'un élément à l'autre. Pour chaque entrée d'une table, les colonnes indiquent :

- les éléments morphologiques constitutifs de l'entrée décrite (c'est-à-dire, les pronoms clitiques figés avec le verbe : Ppv =: se figé, Ppv =: y figé, etc.) ;
- les constructions additionnelles à la construction de base ;
- les propriétés distributionnelles des arguments (catégorie grammaticale : N0 =: Qu P, valeur de la préposition : Prép2 =: de, trait sémantique : N1 =: Nhum, etc.) ;
- les propriétés transformationnelles (redistribution : [passif par], pronominalisation : Ppv =: y, etc.).

19. Notons que nous avons également corrigé les fautes d'orthographe des tables AD à IS2 de G. Gross qui étaient déjà en version Excel.

Ainsi, dans la table 9, une colonne intitulée Qu Psubj = de V2-inf W permet de coder si un verbe appartenant à cette table autorise que son complément direct (de position 1, comme l'indique la colonne N1 =: Qu Psubj) soit une infinitive introduite par la préposition *de* et contrôlée par N2 (*Luc a (ordonné+dit) à Zoé* **de chanter**, à contraster avec **Luc a dissimulé à Zoé de chanter*). Rappelons que la construction de base de la table 9 est N0 V N1 à N2, qui accompagnée de la colonne N1 =: Qu Psubj et Qu Psubj = de V2-inf W revient à admettre la construction N0 V de V2-inf W à N2.

Une autre colonne intitulée Qu P = Aux V0-inf W permet de coder si un verbe appartenant à la table 9 autorise que son complément direct (comme l'indique N1 =: Qu Pind) soit une infinitive directe à un temps composé et contrôlée par le sujet N0 (*Luc a (dit+dissimulé) à Zoé* **avoir chanté**, à contraster avec **Luc a ordonné à Zoé avoir chanté*).

On a donc ici une structure hiérarchique implicite entre colonnes. En effet, la colonne Qu Psubj = de V2-inf W dépend d'une colonne qui indique que la complétive est au subjonctif (*Luc a (dit+ordonné) à Léa que Zoé parte demain*) tandis que la colonne Qu P = Aux V0-inf W dépend d'une colonne qui indique que la complétive est à l'indicatif (*Luc a (dit+dissimulé) à Léa que Zoé part demain*).

Une propriété peut également servir de renvoi à un emploi qui figure dans une autre table. Ainsi, comme nous l'avons déjà mentionné en 5.2.3, la construction N0 V N2 (E+de N1) dans la table à construction standard 38LD indique pour certains verbes l'existence d'une construction croisée correspondante, laquelle figure dans une autre table de type 37 (ayant comme propriétés définitoires N0 V N1 Prép N2 et Prép2 =: de) :

Luc saupoudre du sel sur les frites (38LD)
Luc saupoudre les frites de sel (37M4)

5.3.2 Propriétés homogènes et plus précises

Les propriétés syntaxico-sémantiques sont représentées par des intitulés succincts (cf. 5.2.1), moins précis que les formalismes utilisés par les analyseurs syntaxiques et les grammaires pour représenter les constructions syntaxiques. Par exemple, dans l'intitulé N0 V vers N, qui représente une construction illustrée par *Des animaux divaguent vers le fleuve*, le symbole N représente un groupe nominal, déterminant compris (par exemple, *le fleuve*). Dans l'intitulé N0 V N1 Dnum N, qui représente la construction de *Luc loue son studio 400 euros*, le même symbole N représente cette fois-ci un substantif, alors que le déterminant (ici *400*) est symbolisé séparément par Dnum. Certaines informations sont donc implicites, non entièrement spécifiées ou représentées de façon non uniforme (Hathout et Namer, 1997).

Un travail systématique sur les intitulés de propriétés a été engagé au LIGM, afin d'élever légèrement leur degré de formalisation, sans toutefois en modifier substan-

tiellement les conventions d'origine, afin de garder leur compacité et leur lisibilité [20]. L'objectif est de rendre l'ensemble des classes cohérent.

Tout d'abord, certains intitulés de colonnes peuvent être différents d'une table à l'autre alors qu'ils dénotent la même propriété linguistique. Après les avoir repérés, nous avons choisi une notation commune et effectué les transformations nécessaires.

Certaines différences sont dues simplement à des erreurs d'inattention, ou à des détails qui n'ont pas été comparés à l'existant. C'est ainsi que l'intitulé Det =: E a été remplacé par Det =: <E> pour être identique aux autres tables de noms (cela concerne les tables AN01, ANDN et F1A à F9, cf. Tab. 5.1(b)).

D'autre part, l'intitulé [extrap] présent dans les tables de M. Gross et l'intitulé il V N0 W utilisé par BGL ont la même signification (cf. Tab. 5.1(a)). Nous avons donc renommé les intitulés de toutes ces colonnes en [extrap].

Il fallait cependant veiller à ne pas abuser de ce type de regroupement pour ne pas perdre une information sous-jacente. Ainsi, les deux intitulés [passif par] et N1 être Vpp par N0 peuvent paraître identiques, alors qu'avec le premier les autres objets sont conservés, ce qui n'est pas le cas avec le deuxième. L'intitulé N1 être Vpp, désignant le passif sans compléments, est assez proche également de N1 être Vpp par N0, alors qu'il a la particularité d'être acceptable au présent et d'être interprété comme un résultat statique du procès.

De plus, certains intitulés de colonnes laissaient implicites des informations pourtant nécessaires à leur exploitation automatique. Ainsi, l'intitulé [pc z.] («Prép ce = zéro») signifie que la préposition et le *ce* de la complétive (s'il est présent) peuvent être effacés. Le problème est que cet intitulé ne précise pas quel est l'argument concerné par le fait de pouvoir prendre la forme Qu P, en plus de la forme Prép ce Qu P. Par exemple, dans la table 16 de M. Gross, les deux arguments sont des complétives (sa construction de base est N0 V Prép N1 Prép N2). Nous avons créé les intitulés Prép N1 =: Prép ce Qu P = Qu P et Prép N2 =: Prép ce Qu P = Qu P pour expliciter le fait que la propriété concerne l'argument N1 ou N2 [21].

Les tables de verbes locatifs contenaient une colonne source/destination, qui ne tenait pas compte des différents arguments présents dans chaque table. C'est pourquoi elle a été remplacée par N0 V Loc N1 source Loc N2 destination pour la table 35L, par N0 V Loc N1 Loc N2 pour la table 35ST et par N0 V N1 Loc N2 source Loc N3 destination pour les tables 32PL, 38LH et 38LS [22]. Cette clarification a entraîné l'éclatement de ces tables (voir 6.2.3), afin de séparer celles contenant systématiquement deux arguments locatifs de celles n'en contenant qu'un seul.

Un dernier exemple, la propriété (N1)(de V1 W) codée dans la table 12 de M. Gross

20. En ce qui concerne les adverbes, nous avons de plus scindé deux ressources ne suivant pas les mêmes principes de représentation, ce qui sera détaillé en 6.5.1.

21. Nous avons réalisé ce travail en collaboration avec Eric Laporte en 2007-2008 (Tolone, 2009). Nous avons repris la même notation en 2009 pour les tables de noms ANA de J. Labelle, APE2, APE3 et APP2 de R. Vivès (cf. Tab. 5.1(b)).

22. Nous avons réalisé ce travail en collaboration avec Eric Laporte en 2008.

a été réintitulée Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W). Cette propriété relie la construction illustrée par *Le ressort empêche la bague de glisser*, à celle de *Le ressort empêche que la bague glisse*. L'emploi du symbole N1 pour désigner le sujet qui subit la montée, ici *la bague*, était critiquable car ce symbole désigne déjà par ailleurs l'ensemble de la complétive objet, en l'occurrence *que la bague glisse*. C'est pourquoi il a été remplacé par Ni [23].

De plus, certaines colonnes étaient restées sans intitulé :

– la table 39 de BGL (cf. Tab. 5.1(a)) possédait une colonne sans intitulé. Elle a été supprimée car un seul verbe était concerné et sa signification n'a pas pu être rétablie ;
– la table 35RR de BGL (cf. Tab. 5.1(a)) n'était au départ qu'un projet de table initié par A. Guillet et douze intitulés de colonnes n'ont pas été retrouvés.
  Nous avons pu reconstituer la plupart des intitulés en partant du codage des entrées, sachant que la construction de base de la table 35RR est N0 V Prép N1 Prép N2 [24]. Les 2 premières colonnes ont été codées N0 =: Nhum et N0 =: N0-hum, les colonnes 5 et 6, N1 =: Nhum et N1 =: N0-hum et les colonnes 9 et 10, N2 =: Nhum et N2 =: N0-hum, ce qui semble assez cohérent puisque toutes les tables codent cette information. Les colonnes 7 et 12 ont été codées N1 = : le fait Qu P et N2 = : le fait Qu P et la colonne 11, N2 =: Dnum Nmes. La colonne 3 a été codée N0 et N2 V Prép N1 (*Max conspire avec Luc contre le président = Max et Luc conspirent contre le président*). En ce qui concerne les colonnes 4 et 8, plusieurs propositions ont été faites, mais sans faire l'unanimité, elles restent donc sans intitulé ;
– quelques tables d'expressions figées sont également dans ce cas : E1HC (sept intitulés), E1 (trois intitulés), EAPC (deux intitulés), EPDETC (un intitulé). Elles ont été créées par M. Gross (cf. Tab. 5.1(c)), qui n'a pas eu le temps des les achever ni de les publier pour certaines, et il est difficile de retrouver les intitulés.

5.3.3 Documentation des propriétés

Les propriétés syntaxico-sémantiques ne sont pas définies avec précision par leurs intitulés. Elles sont documentées dans des publications scientifiques mais cela reste insuffisant :

– toutes ne sont pas documentées, comme c'est le cas pour certaines classes d'expressions figées de M. Gross (cf. Tab. 5.1(c)) qui n'ont même pas été publiées ;
– leur documentation est parfois difficilement accessible, car certains ouvrages sont moins diffusés que d'autres. C'est le cas des rapports ou des thèses n'ayant pas débouché sur une publication : Boons *et al.* (1976a) pour les verbes, Meunier (1981) et Giry-Schneider et Balibar-Mrabti (1993) pour les noms prédicatifs ;
– aucun ouvrage n'a été traduit en anglais ;

23. Nous avons réalisé ce travail en collaboration avec Eric Laporte, Christian Leclère et Takuya Namakura en 2009 (Laporte, 2010).

24. Nous avons réalisé ce travail en collaboration avec Eric Laporte, Christian Leclère et Benoît Sagot en 2008.

- les définitions manquent de précisions pour les spécialistes d'analyse syntaxique, qui ne sont pas toujours des spécialistes de syntaxe ;
- un même intitulé peut avoir différentes interprétations et représenter une propriété linguistique différente en fonction des classes ; ainsi N0 =: N-hum indique que le sujet N0 de la construction de base peut être occupé par un groupe nominal dénotant une entité non humaine, le verbe conservant son sens canonique (cf. 5.2.2 avec *Le chemisier blouse*), sauf dans la classe 31H où ce même intitulé indique que la phrase prend alors un sens métaphorique, comme dans *Le paysage sommeille*, à contraster avec *Luc sommeille* ;
- deux intitulés similaires peuvent avoir une signification différente dans deux tables distinctes, ou éventuellement dans une même table. Ainsi, la table 36DT comporte essentiellement des verbes prenant un objet direct non humain concret. La propriété N1 =: Nhum y a deux rôles : d'une part elle marque la possibilité pour certains verbes de produire des métaphores (*Paul emprunte une secrétaire au patron*, par rapport à *Paul emprunte cent francs au patron*) ; d'autre part, elle note une sous-classe particulière de constructions où l'objet échangé est strictement humain (*Paul délègue sa secrétaire au patron*). Ce dernier cas est séparé de l'autre par le codage − de la colonne N1 =: N-hum.

L'interprétation de certains intitulés peut donc être difficile (Hathout et Namer, 1997). Pour remédier à ce problème, la documentation des propriétés la plus complète, qui est celle des verbes locatifs (Guillet et Leclère, 1992) (pp. 409-430) a été entièrement revue, étendue à toutes les propriétés des verbes distributionnels, et traduite en anglais [25].

De plus, cela a permis de vérifier pour toutes les classes de verbes à quelle signification chaque intitulé faisait référence, l'objectif étant qu'un intitulé dénote une seule propriété linguistique, qui elle-même n'est désignée que par un seul intitulé dans l'ensemble des tables. Elle est à présent complète et mise à jour dès qu'une modification a lieu dans une table. Elle est incluse dans la version 3.3 et dans l'annexe E.

5.4 Tables des classes

Comme nous l'avons évoqué en 5.2.2, les tables ne codent pas explicitement leurs propriétés définitoires. C'est un problème important pour leur exploitation automatique, puisque ces propriétés sont parmi les plus importantes. Les critères de découpage en classes et les propriétés définitoires ne sont décrites que dans la littérature associée aux tables. Ce constat a motivé depuis quelques années le développement au LIGM d'un nouveau type de tables, appelées *tables des classes* (Constant et Tolone, 2010).

La notion de table des classes a été définie suivant Paumier (2003) [26]. Son rôle est d'assigner des propriétés syntaxico-sémantiques à une classe quand cela est possible,

25. Nous avons réalisé ce travail en collaboration avec Eric Laporte et Christian Leclère en 2008-2011.

26. Elle était appelée au départ *super-table* et elle comportait quelques variantes puisque son objectif était de l'utiliser avec le logiciel Unitex afin de générer un graphe par table à partir d'un graphe générique faisant appel à cette super-table, puis de créer un graphe lexical pour chaque entrée à partir de ces graphes et des tables elles-mêmes.

c'est-à-dire, quand leur valeur est constante pour toute une classe (par exemple, les propriétés définitoires d'une classe).

Ces tables sont au nombre d'une par catégorie grammaticale, donc 4 au total : une pour les verbes, une pour les noms prédicatifs, une pour les expressions figées et une pour les adverbes (cf. Tab. 5.1). Une table des classes regroupe en lignes l'ensemble de toutes les propriétés syntaxico-sémantiques répertoriées pour la catégorie concernée, et liste en colonnes l'ensemble des classes définies pour cette même catégorie. À l'intersection d'une ligne et d'une colonne, le signe + (respectivement, −) indique que la propriété correspondante est vérifiée (respectivement, non vérifiée) par tous les éléments de la classe (c'est-à-dire par toutes les entrées de la table correspondante). Le signe o indique que la propriété est explicitement codée dans la table concernée, car elle est vérifiée par certaines de ses entrées mais pas toutes[27]. Le signe O indique que la propriété n'est pas codée dans la table concernée mais devrait l'être, car elle dépend des entrées. Enfin, le signe ? indique une cellule non encore renseignée.

Par exemple, la table des classes des verbes regroupe les 67 classes de verbes distributionnels et l'ensemble des 551 propriétés syntaxico-sémantiques. Un extrait de cette table est donné à la Fig. 5.2[28]. La notation V_ indique qu'il s'agit de classes de verbes. Dans cette table, on peut voir la construction de base de la classe 33 (cf. section 3.2) : la construction N0 V à N1 est codée + car elle vaut + pour l'ensemble des entrées de la table 33 ; de même pour la classe 1, qui a la construction N0 V Prép V0-inf W codée +.

Les propriétés distributionnelles N1 =: Nhum et N1 =: N-hum se voient assigner o pour la table 33 parce qu'elles dépendent des entrées lexicales. En revanche, pour la table 32H, la propriété distributionnelle N1 =: Nhum est codée + alors que la propriété distributionnelle N1 =: N-hum est codée − car l'objet est obligatoirement humain. Les deux propriétés distributionnelles sont donc définitoires de la table, la deuxième étant constante − pour la table.

Remarquons que le codage − dans la table des classes désigne réellement dans ce cas une propriété que n'accepte aucune des entrées. Dans d'autres cas, il peut désigner simplement une information non pertinente (par exemple, une propriété de N1 pour une entrée qui n'a pas de N1). Cela vient du fait que certaines propriétés sont spécifiques à une table donnée. Elles ne seraient pas pertinentes pour une autre ; ainsi, N2 bénéficiaire ne figure qu'en 36DT. L'absence d'une propriété dans une table peut donc signifier qu'elle est constante + ou − pour toutes les entrées de cette table, ou qu'elle n'a pas été jugée intéressante, ou encore qu'elle n'est pas pertinente pour la table en question.

La table des classes fait apparaître les propriétés définitoires, toutes celles qui sont

27. Dans Paumier (2001), on faisait référence à la propriété codée dans la table par une variable : @A pour la première colonne contenant la première propriété à droite, @B pour la deuxième, etc. De plus, on pouvait utiliser la négation d'une propriété avec la variable !@A pour récupérer le codage inverse sans devoir créer une colonne le contenant.

28. La section 9.3 contient un autre extrait de cette même table des classes, avec l'ensemble des propriétés pertinentes pour la classe 31R, toutes ne figurant pas ici.

Propriété \ table	V_1	V_2	V_4	V_31R	V_31H	V_33	V_32H
N0 =: Nhum	o	+	−	o	+	o	o
N0 =: N-hum	−	−	−	o	−	o	−
N0 =: Nnc	o	−	−	−	−	−	−
N0 =: Nnr	−	−	+	−	−	o	o
N0 =: V1-inf W	−	−	+	−	−	−	−
<ENT>V	o	o	o	o	o	o	o
Ppv =: se figé	o	o	−	o	o	o	o
N0 V	−	−	o	+	+	o	−
N0 V N1	−	−	+	−	−	−	+
Prép1	o	−	−	−	−	−	−
N0 V à N1	−	−	−	−	−	+	−
N1 =: Nhum	−	−	o	−	−	o	+
N1 =: N-hum	−	+	o	−	−	o	−
N1 =: Qu P	−	−	−	−	−	−	−
N1 =: Qu Psubj	−	−	−	−	−	−	−
N0 V Prép N1 V0-inf W	−	o	−	−	−	−	−
N0 V N1 V0-inf W	−	o	−	−	−	−	−
N0 V V0-inf W	−	+	−	−	−	−	−
N0 V Prép V0-inf	+	−	−	−	−	−	−
N0 V Prép Nhum	o	−	−	−	−	−	−
N0 V Prép N-hum	o	−	−	−	−	−	−
N0 V Nhum	o	−	−	−	−	−	−
N0 V N-hum	o	−	−	−	−	−	−

TABLE 5.2: Extrait de la table des classes des verbes distributionnels

constantes + ont été codées, comme nous le verrons en détail au chapitre 6. Notons que dès que l'on peut coder une information avec différentes combinaisons de propriétés, un choix arbitraire a été fait et les autres possibilités ont été codées − dans la table des classes pour ne pas engendrer de redondance. Par exemple, pour la table 9, la construction de base est N0 V N1 à N2, mais aurait pu également être N0 V N1 Prép N2 avec Prép2 =: à définitoire, mais l'on a préféré intégrer la préposition dans la construction de base lorsqu'une seule était possible. Les deux propriétés N0 V N1 Prép N2 et Prép2 =: à sont donc codées − dans la table des classes.

La table des classes permet également de coder toutes les propriétés syntaxico-sémantiques pour chaque table, y compris celles dont on ne trouve la description que dans la littérature et qui de ce fait, ne sont pas exploitables alors qu'elles peuvent être pertinentes. C'est le cas, par exemple, des propriétés transformationnelles [passif par] et [passif de], qui sont fréquentes en français. Ces propriétés fondamentales ont en général été étudiées et décrites dans les thèses, ce qui signifie que lorsqu'elles ne sont pas codées dans une table, elles sont constante + ou − (avec peut-être quelques exceptions répertoriées dans les thèses), sans pour autant être définitoire puisqu'elles n'ont pas participées au découpage de la classe en question. Cet ensemble de propriétés est à coder prioritairement dans la table des classes, mais il n'est pas toujours facile à repérer.

En outre, le choix des propriétés codées dans chaque table ayant été arbitraire, certaines propriétés peuvent, après réflexion, se révéler pertinentes, soit constante + pour la table, soit variant selon les entrées. Dans ce dernier cas, cela signifie qu'il s'agit d'une propriété qui devrait être codée o dans la table des classes et codée dans la table. En attendant d'être effectivement codée dans la table, elle est codée O dans la table des classes.

La construction de ces tables des classes a permis une homogénéisation importante des tables et en particulier des intitulés de colonne. C'est ainsi que de nombreuses colonnes ont changé de nom (cf. 5.3.2), cela a permis également de revoir la notation des constructions de base de chaque table, ce qui sera détaillé au chapitre 6.

5.5 Codage des tables

Avant de détailler le codage restant à faire (5.5.2), nous abordons le problème des informations manquantes dans les tables (5.5.1). Ensuite, nous évoquons les difficultés que le codage peut poser (5.5.3).

5.5.1 Informations manquantes

Comme cela a été signalé dans Gardent *et al.* (2006), certaines informations sont incomplètes ou absentes des tables du Lexique-Grammaire, comme par exemple les fonctions grammaticales et les rôles thématiques remplis par chacun des arguments. Mais d'autres propriétés, qui ne sont généralement pas utilisées par les analyseurs, sont présentes, comme l'interprétation temporelle des infinitives, laquelle peut être utile à la construction d'une représentation du sens du texte.

Comme cela a été expliqué dans Laporte (2010), les fonctions grammaticales ne sont pas toutes codées, car elles recouvrent des propriétés syntaxico-sémantiques, généralement plus factuelles, avec lesquelles elles font en partie double emploi. Ainsi, la notion de complément d'objet direct se fonde sur différentes propriétés qui ne coïncident pas toujours : position après le verbe, absence de préposition, pronominalisation, passivation, etc. (Gross, 1969). Ce sont plutôt ces propriétés qui sont codées dans les tables du Lexique-Grammaire, ce qui a l'avantage d'être plus précis. En particulier, les auteurs des tables du Lexique-Grammaire des verbes distributionnels ont joué un rôle pionnier dans la délimitation entre les compléments essentiels (objets) et circonstanciels (adjoints, modifieurs). Ainsi, ils ont décrit comme complément essentiel le complément direct des verbes de la classe 32NM (*Luc chausse une grande taille* / *La pièce sent le jasmin*), souvent considéré comme circonstanciel. Il en est de même du complément indirect de nombreux verbes locatifs (*Luc place sa voiture contre le mur*, table 38LD). Ils ont également recensé de nombreux compléments qui ont un comportement intermédiaire entre ceux d'un complément essentiel et d'un complément circonstanciel, par exemple *sur ce point* dans *Luc se ravise sur ce point* (table 31H).

En ce qui concerne les rôles thématiques et plus généralement la formalisation du sens, les auteurs des tables du Lexique-Grammaire se sont limités à des phénomènes dont ils ont pu encadrer l'observation par des tests syntaxiques (voir 5.5.3).

5.5.2 Codage inachevé

Le codage des tables n'est pas terminé.

En effet, du côté des propriétés :

- les ? dans la table des classes correspondent pour la plupart à des propriétés non encore étudiées dans certaines tables. L'importance de certaines d'entre elles empêche l'analyse des constructions correspondantes par un analyseur ;
- les constructions dont la formation est régulière, comme la négation ou les propositions relatives, ont été négligées, sauf lorsqu'elles varient en fonction des éléments lexicaux ;
- certaines constructions, notamment pronominales, ne sont pas codées.

De même, du côté des entrées :

- certaines entrées dans les tables des verbes n'ont pas encore été codées et contiennent des signes $\sim$ à la place des $+$ et des $-$. Ces emplois de verbes ne seront pas pris en compte dans un analyseur, alors qu'ils sont parfois essentiels ;
- d'autres entrées sont tout simplement manquantes tout n'ayant pas été encore étudié, surtout dans certaines catégories.

Ceci est le cas des adjectifs prédicatifs, pour lesquels un travail de découpage est en cours d'achèvement par J. Giry-Schneider (cf. Tab. 5.1(e)). Ce travail a permis d'établir 38 classes, mais il reste encore à coder un ensemble de propriétés syntaxico-sémantiques (à définir) pour les entrées retenues. Il faut toutefois remarquer que les tables Adj1 à Adj13 de L. Picabia existent en version 1, cependant leur découpage n'étant pas assez fin, il a été décidé par le LIGM qu'il était préférable d'en créer de nouvelles. Certains adjectifs non prédicatifs ont également donnés lieu à une étude par E. Laporte et J. Giry-Schneider

(cf. Tab. 5.1(f)), mais seulement trois tables existent au format Excel, dont deux doivent être fusionnées, c'est pourquoi elles ne sont pas prises en compte actuellement.

Ajoutons enfin que les 5 tables de noms FD1 à FD4 de L. Pivaut (cf. Tab. 5.1(b)) viennent d'être numérisées, ce qui demande un travail de correction manuelle avant d'envisager leur utilisation. De même, les 3 tables d'expressions figées ZD, ZM et Z de L. Danlos (cf. Tab. 5.1(c)) n'ont pas été numérisées car il existe une version plus récente et plus complète avec un découpage différent : 24 tables au lieu de 8, mais toutes n'ont pas été retrouvées (voir 6.4.4).

Toutes ces informations sont certainement indispensables au bon fonctionnement des analyseurs syntaxiques symboliques. Malgré ces limitations, il est difficile de contester la richesse des informations contenues dans les tables du Lexique-Grammaire, en comparaison avec d'autres lexiques structurés. Le balayage du lexique et le recensement des constructions sont considérables. La délimitation systématique entre constructions figées et constructions libres est difficile à trouver ailleurs, si ce n'est dans les tables du Lexique-Grammaire d'autres langues.

En ce qui concerne les verbes distributionnels, nous avons mis à jour les entrées. Certains choix avaient été fait à une époque où les données linguistiques étaient répertoriées sur des fiches cartonnées, mais ils n'avaient jamais été répertoriés dans les versions électroniques ultérieures. De plus, un index électronique recense toutes les entrées et indique les tables dans lesquelles elles apparaissent, en donnant plusieurs exemples de phrases. Nous avons donc mis en correspondance les tables et cet index, tout en tenant compte des fiches cartonnées[29].

Cependant, le découpage des verbes très fréquents, et pour lesquels il est difficile d'identifier clairement tous les sens, avait parfois été mis de côté. Une de nos priorités a été de les ajouter. Ces verbes ont donc été entièrement passés en revue pour que les tables soient les plus complètes possible. En revanche, ces verbes n'ont pas encore été codés, ce qui pose un réel problème pour leur reconnaissance dans les textes.

Afin d'illustrer ce problème, une liste de 30 verbes (en prenant en compte les formes pronominales, les pronoms clitiques figés, les adverbes obligatoires et autres) a été établie, tous étant considérés comme des verbes fréquemment représentés dans les textes. La Tab. 5.3 montre le nombre d'entrées codées et non codées pour ces verbes. On remarque qu'en moyenne les entrées non codées constituent la moitié de ces verbes.

C'est ainsi que l'on s'aperçoit que le verbe *rivaliser* a une entrée qui est codée dans la table 35RR de BGL, dont l'exemple est *Max rivalise de générosité avec Luc*. Or, il figure également sans être codé, dans la table 35S de BGL qui reconnaît *Les sculptures rivalisent avec les Mayol*, avec un seul argument, ce qui est le sens le plus courant.

De même, le verbe *contester* a une entrée codée dans la table 6 de M. Gross et illustrée par *Max conteste qu'il soit responsable*, et une autre entrée dans la table 9 de M. Gross

29. Nous avons réalisé ce travail en collaboration avec Christian Leclère en 2008-2009 (Tolone, 2009). Une comparaison des entrées verbales entre la première version de 2008 et la version 3.3 est réalisée à la section 9.2.

Verbes	Entrées codées	Entrées non codées
apporter	4	1
avoir	14	5
bouger	1	9
contester	2	2
craindre	3	5
doser	2	2
être	19	8
faire	15	14
gagner	8	5
jouer	26	15
laisser	5	13
lire	4	5
manger	4	5
monter	14	17
passer	14	27
perdre	13	9
porter	12	8
prendre	15	18
préparer	4	3
rattraper	6	5
revenir	15	6
rivaliser	1	1
sembler	0	1
sentir	4	6
servir	8	6
suivre	9	12
susciter	0	2
tirer	11	21
trouver	6	9
venir	11	6
vouloir	5	2
`total`	255	260

TABLE 5.3: Nombres d'occurrences d'entrées codées et non codées pour une liste de verbes fréquents

avec comme exemple *Max conteste à Luc qu'il puisse occuper ce lieu*. En revanche, deux autres entrées ne sont pas codées, l'une dans la table 32H de BGL qui reconnaît *Max conteste le directeur*, et l'autre dans la table 36DT avec pour exemple *Max conteste à Jean (le+la possession du) ballon*. Ce qui pose problème, puisque les deux uniques tables dont les entrées sont codées mettent en jeu des complétives.

Enfin, le verbe *susciter* ne possède aucune entrée codée, alors qu'il figure dans la table 36DT de BGL avec comme exemple *Que Léa boive suscite des craintes à Max*, ainsi que dans la table 38R de BGL qui reconnaît *Ceci a suscité une (vive réplique+réaction)*, il ne peut donc être reconnu dans les textes. De plus, il pourrait également figurer en tant que verbe support de l'entrée *espoir* dans la table APE3 de noms prédicatifs, qui a comme exemple *Luc a de l'espoir (E+en Max+en l'avenir)*. Pour cela, il suffirait d'y ajouter une construction permettant de reconnaître l'exemple *(Max+L'avenir) suscite l'espoir de Luc*, comme c'est le cas du verbe *donner*. En effet, la construction N1nr donner Det N à N0 est codée dans la table et reconnaît *(Max+L'avenir) donne de l'espoir à Luc*.

Notons que, même dans un lexique le plus complet possible, il est impossible de couvrir toutes les entrées car on ne peut pas établir de liste définitive, puisque de nouveaux mots apparaissent régulièrement. Dans les tables du Lexique-Grammaire, les verbes techniques ont été écartés, pour les raisons suivantes : ils sont très nombreux, ils sont plus rares dans les textes, ce qui les rend moins prioritaires. De plus, leur emploi est peu connu et donc difficiles à coder sans être spécialiste du domaine concerné. Malgré cela, une liste a été établie mais uniquement dans des fiches cartonnées.

5.5.3 Difficulté du codage

Nous discutons de la difficulté que représente le codage des tables du Lexique-Grammaire puisqu'il peut y avoir des erreurs ou des différences d'acceptabilité entre linguistes. Ceci peut aller jusqu'à remettre en cause sa validité, alors qu'il se prononce sur des points relativement vérifiables. Nous évoquons la possibilité d'envisager l'utilisation d'un corpus pour attester certaines phrases mais sans exhaustivité. Enfin, nous abordons la signification du signe $-$.

Validité du codage

Certaines informations contenues dans les tables du Lexique-Grammaire peuvent être incorrectes, ainsi que cela a été signalé dans Gardent *et al.* (2006). En effet, plusieurs sources d'erreurs expliquent la présence d'informations invalides.

Tout d'abord, il existe des erreurs matérielles. Des anomalies dans les programmes de gestion des tables ont inversé tous les signes $+$ et $-$ dans certaines entrées, par exemple *traîner là* dans la table 1 (entrée qui par la suite a été supprimée de table 1 des verbes et ajoutée dans la table CADV des expressions figées). Nous avons nous-même introduit, au cours de la révision des intitulés (cf. 5.3.2), plusieurs erreurs qui ont été corrigées par la suite.

Ensuite, certains verbes supports ont été codés dans les tables de verbes distributionnels. Ainsi, le verbe *faire* dans *Max fait (du sport+du tennis+de la nage)* est codé dans

la table 32R3. Or, la table FD2 de L. Pivaut de noms prédicatifs décrit les noms *sport* et *tennis*, ainsi que d'autres noms dans le même cas. De même, la table FD4 de L. Pivaut décrit le nom *nage*. Il faudrait donc supprimer tous les verbes supports contenus dans les tables de verbes distributionnels, après s'être assuré que tous les noms prédicatifs correspondants sont bien codés dans les tables de noms prédicatifs.

On pourrait faire de même pour le verbe *subir* dans *Le pétrole subit une hausse* codé dans la table 32NM. Or, il manque certains noms prédicatifs, par exemple, le nom *hausse* ne figure pas dans les tables de noms prédicatifs.

Il faut cependant veiller à ne pas le faire abusivement. Ainsi, le verbe *avoir* dans *Max a une cible* est codé dans la table 32R3 et le nom *cible* figure dans la table AN10 de A. Meunier, mais avec un sens différent : dans la table 32R3, il s'agit du sens *Max a eu la cible (= il ne l'a pas ratée)*, alors que dans la table AN10, c'est le sens *Max cible quelque chose*.

Comme cela a déjà été mentionné dans Laporte (2010), la présence d'erreurs dans les tables du Lexique-Grammaire ne doit pas nous faire oublier ses points forts en ce qui concerne la validité. Les tables se prononcent explicitement sur des points vérifiables et restent assez neutres par rapport aux différentes théories syntaxiques. Ses auteurs se sont concentrés sur des phénomènes relativement vérifiables, c'est-à-dire ceux pour lesquels l'observation est aisément reproductible. Ils ont ainsi marqué l'aspect *processif* ou *statique* de certaines constructions, comme N2 V N1, illustrée par *Le rideau cache le sac* (statique), que nous pouvons contraster avec la construction de base de la même entrée : *Luc cache le sac derrière le rideau* (processif). Dans le cas de la construction N1 V Loc N2, le marquage de ce trait sémantique n'a pas été jugé suffisamment reproductible pour être systématisé : si l'aspect est nettement processif dans *Le volet claque contre le mur*, et nettement statique dans *Le carton tient contre la caisse*, l'intuition sémantique est moins claire dans *Le frein frotte sur la jante*. D'une manière générale, les auteurs des tables du Lexique-Grammaire se sont entourés de précautions méthodologiques en vue d'assurer la reproductibilité de leurs observations (Laporte, 2008), et le recours à l'intuition y est plus sévèrement encadré que, par exemple, dans Levin (1993). Il s'en suit une base empirique plus solide.

Il arrive qu'un cadre théorique ait une difficulté à prendre en compte un fait observé dans les tables du Lexique-Grammaire : c'est probablement ce que Hathout et Namer (1997) (p. 5) entendent par «certaines transformations sont linguistiquement incorrectes, dans le cadre théorique considéré» (HPSG), ce qui est illustré par la construction N1 se V auprès de N3hum de ce Qu P (*Luc se réjouit auprès de Marie de ce que le film sorte*). Cette construction à trois arguments, qui dénote un acte de parole, coexiste avec une construction à deux arguments (*Que le film sorte réjouit Luc*, construction de base de la table 4). Dans cette construction, on ne peut pas toujours considérer que le troisième argument, formellement absent, est en fait sémantiquement présent. Une telle situation est une anomalie par rapport à la plupart des théories actuelles.

L'incorrection en fait se situe plutôt du côté du cadre théorique que du phénomène observé. La neutralité par rapport aux théories syntaxiques explique par ailleurs le choix d'un degré de formalisation limité (cf. 5.3.2). Un formalisme plus complexe est

nécessairement plus dépendant d'une théorie, ce qui peut gêner l'observation éventuelle de faits auxquels cette théorie n'a pas été adaptée.

Utilisation d'un corpus

Remarquons que les différents auteurs des tables du Lexique-Grammaire peuvent avoir des différences d'interprétation et surtout une rigueur variable. Seulement deux codages sont possibles (+/−) (Harris, 1952). On peut être laxiste pour les + (c'est-à-dire permettre des phrases presque inacceptables, ce qui permet d'être plus couvrant mais augmente l'ambiguïté) ou au contraire trop intransigeant pour les − (ce qui engendre des difficultés puisque ces formes ne pourront jamais être reconnues). Ceci constitue le problème du codage binaire. On peut tout de même noter que, pour un analyseur syntaxique, il est préférable de couvrir plus pour reconnaître même les phrases rares. Le codage aurait pu être de la forme ++/+/?/−/−−, mais il aurait été plus difficile d'avoir des données cohérentes d'un auteur à l'autre.

Dans les faits, les auteurs des tables du Lexique-Grammaire ont cherché à infléchir légèrement leurs jugements d'acceptabilité dans le sens de la tolérance. En effet, ils n'ont pas cherché à appuyer leurs décisions sur des attestations dans des corpus car cela était irréalisable à l'époque (Boons *et al.*, 1976b) (p. 37). Un contrôle de la validité plus objectif aurait été lourd et aurait sans doute compromis la faisabilité du projet. La grande couverture en informations a été préférée à l'objectivité.

En effet, dans la plupart des cas, il suffit qu'un exemple soit trouvé, pour que la propriété soit considérée comme acceptée, même si elle ne l'est que dans certaines phrases ou dans certains contextes. Par exemple, N0 lieu de destination de N1 est noté + pour le verbe *hisser* (table 38L), bien qu'on puisse imaginer toutes sortes de situations dans lesquelles, si Max hisse une caisse, il ne soit pas le point d'arrivée de la caisse.

Aujourd'hui, confronter les tables du Lexique-Grammaire avec un corpus serait un travail intéressant. On pourrait en effet, rendre le codage moins binaire en indiquant une probabilité d'apparition de chaque construction pour chaque entrée après avoir appliqué un analyseur syntaxique sur un corpus de taille conséquente.

En revanche, il ne serait pas possible de relier à des attestations toutes les informations contenues dans les tables du Lexique-Grammaire. Elles représentent, pour les verbes, un balayage du lexique (environ 13 800 entrées verbales, même si toutes n'ont pas encore été codées) croisé avec un balayage d'environ 500 propriétés syntaxico-sémantiques, au cours duquel les mêmes constructions sont testées aussi bien sur les entrées rares (par exemple, *godailler*) que sur les entrées fréquentes (par exemple, *bouillir*). Un corpus représente également un balayage croisé, mais partiel, sans pouvoir garantir que la totalité des combinaisons soit passée en revue. La différence importante est qu'il n'atteste pas d'inacceptabilités. Le choix de couvrir une grande masse d'informations justifie d'ailleurs en partie aussi celui d'un degré de formalisation limité (cf. 5.3.2).

Signification du signe −

Une notation différente pourrait être envisagée pour la table des classes. En effet, comme nous l'avons mentionné précédemment, tous les signes − n'ont pas la même valeur, mais il n'est pas toujours évident de les distinguer rigoureusement :

– certaines propriétés ont une vraie valeur − car elles ne sont acceptées par aucune entrée de la table (par exemple, N1 destination pour la table 2) ;
– d'autres sont codées − lorsqu'elles ne sont pas pertinentes car elles dénotent un élément qui est inapproprié pour la table (par exemple, N1 destination pour la table 31H, qui a comme construction de base N0 V et n'a donc pas de complément N1, comparée à la table 2 qui a comme construction de base N0 V Loc N1 V0-inf W).

La distinction a été faite uniquement dans la table des classes des expressions figées pour tester l'utilité de cette notation. Pour cela, nous avons ajouté manuellement un signe / pour coder une propriété non pertinente uniquement dans les cas faciles à repérer comme l'exemple ci-dessus.

Lorsqu'une propriété n'est pas pertinente, c'est qu'il existe d'autres propriétés qui permettent de déduire que cette propriété n'est jamais acceptée. On pourrait formaliser ces dépendances/redondances entre propriétés, avec des formules logiques du type «si aucune construction ne contient N2 alors aucune propriété distributionnelle de l'argument N2 n'est acceptée», ce qui permettrait d'automatiser la distinction entre − et /. Mais toutes les dépendances/redondances entre entrées sont difficiles à décrire. Par exemple, les verbes pronominaux n'ont pas de passif :
Les clients s'arrachent ce modèle
**Ce modèle s'est arraché par les clients*
Mais il est difficile de savoir si c'est une propriété pertinente ou pas car il n'y a pas de critère précis, alors que ce qui sépare + de − est défini sur la base d'un jugement d'acceptabilité.

De plus, l'ajout d'un signe / ne serait qu'une information linguistique supplémentaire qui n'aurait pas de réel impact dans un analyseur. En revanche, c'est une notation qui peut aider à la détection d'erreurs dans les tables. Par exemple, si une propriété codée − pour une entrée est recodée temporairement +, le temps d'analyser un corpus de phrases contenant l'entrée correspondante afin de voir si cela change les résultats de l'analyseur, et si oui, étudier ces résultats. Les propriétés codées avec un signe / ne sont pas concernées et n'ont pas besoin d'être étudiées avec attention par des linguistes qui souhaitent revenir sur certains codages de la table des classes.

5.6 Bilan

Cette section résume rapidement comment les tables du Lexique-Grammaire sont effectivement utilisées dans un processus d'analyse syntaxique symbolique complet (5.6.1), avant de terminer par une analyse des avantages de cette méthode par rapport à l'approche probabiliste (5.6.2).

5.6.1 Explicitation du processus complet d'utilisation des tables

La version actuelle des tables a permis d'envisager une utilisation de ces données lexicales dans des outils de TAL (Tolone, 2009). A cette fin, un lexique syntaxique généré à partir des tables a été développé initialement pour les verbes et les noms (Constant et Tolone, 2010), puis pour toutes les catégories. Il s'agit du lexique *LGLex*, que nous détaillons au chapitre 7.

Ensuite, nous avons pu intégrer une partie de ce lexique dans l'analyseur syntaxique à grande échelle FRMG (cf. section 4.2). Cette intégration a été l'objet du travail dans Sagot et Tolone (2009b); Tolone et Sagot (2011), décrit au chapitre 8, qui a permis la conversion des tables au format Alexina, le format du lexique Le*fff* (cf. section 4.1). L'analyseur syntaxique FRMG couplé à ce lexique a ensuite été évalué sur le corpus de référence de la campagne Passage, comme cela a été initialement fait sur le corpus de référence de la campagne EASy dans Sagot et Tolone (2009a); Tolone et Sagot (2011). Nous verrons cela au chapitre 10.

5.6.2 Conclusion

Les choix faits lors de la construction des tables du Lexique-Grammaire sont motivés par un certain nombres de principes et d'objectifs parfois originaux : un vaste recensement du lexique et des constructions, la priorité donnée aux propriétés factuelles sur les contraintes liées à des théories spécifiques, une exigence de reproductibilité des observations. Ce sont ces caractères qui nous ont permis d'envisager l'exploitation des tables du Lexique-Grammaire dans des systèmes de TAL.

En effet, l'ensemble des travaux récents de récupération et de mise en cohérence de ces données linguistiques a permis d'obtenir pour l'ensemble des catégories représentées une nouvelle version des tables du Lexique-Grammaire, qui, combinée avec les tables des classes, constitue un ensemble complet et synthétique de données linguistiques.

Il est important de noter que, malgré les erreurs et les différences de jugements entre les auteurs des tables du Lexique-Grammaire, aucune approximation n'a été faite. Ces données sont donc aussi correctes que possibles. Bien entendu, les systèmes par acquisition automatique fonctionnent bien et peuvent plus facilement prendre en compte des nouveaux mots. Mais ils comportent aussi beaucoup d'erreurs car ils omettent des phénomènes rares ou absents des corpus d'apprentissage. Il existe des moyens de corriger ces erreurs, notamment au moyen de techniques automatiques, telles que celles décrites dans Sagot et de La Clergerie (2008), mais cela prend du temps et se révèle insuffisant à tout détecter. Remarquons que ces techniques peuvent être également employées pour améliorer les ressources manuelles, en corrigeant des erreurs dans les tables et permettre ainsi de les améliorer.

6

Constructions de base pour toutes les catégories

Les constructions de base ont subi des changements au cours des années alors que ce sont les plus importantes à définir. L'objectif de ce chapitre est de suivre leur trace et de comprendre le sens de leur modification. Nous avons réalisé ce travail pour l'ensemble des catégories, à savoir les verbes distributionnels, les noms prédicatifs, les expressions figées et les adverbes (cf. Tab. 5.1).

Nous listons tout d'abord dans les sections 6.1 et 6.2, les modifications apportées aux constructions de base des classes des verbes distributionnels de M. Gross et de BGL. Puis, dans la section 6.3, nous passons en revues les différentes tables de noms prédicatifs, avec les différents verbes supports associés. Ensuite, dans la section 6.4, nous détaillons le cas des expressions figées et enfin, dans la section 6.5, celui des adverbes, en expliquant comment nous avons scindé deux ressources (adverbes simples et adverbes (semi-)figés) qui ne suivent pas les mêmes principes de représentation. Nous finissons par un bilan sur le nombre d'entrées par catégories et un recensement de l'ensemble des propriétés définitoires vraies pour toutes les entrées d'une table, incluant la construction de base, dans la section 6.6.

Il faut noter que nous avons ajouté, dans la table des classes concernée, toutes les propriétés définitoires vraies pour toutes les entrées d'une table, qui sont définies dans ce qui suit. Elles sont donc prises en compte lors de la conversion vers le lexique *LGLex*.

6.1 Les classes des verbes distributionnels : Les complétives et les infinitives

Les classes des verbes distributionnels de M. Gross (cf. Tab. 5.1(a)) sont au nombre de 19 lors de leur création (Gross, 1975) : de 1 à 3, les classes des infinitives et de 4 à 19, celles des complétives. Nous allons détailler pour chaque table quelle est sa construction de base actuelle et expliquer quelles sont les modifications qui ont été nécessaires, comme cela a été fait dans Tolone (2010).

Une seule table n'a subi aucun changement, il s'agit de la table 16 décrite dans Gross (1975) (p. 208), qui a pour construction de base N0 V Prép N1 Prép N2, où les arguments N1 et N2 peuvent être des complétives (*Flo déduit que Guy est arrivé de ce qu'il y a du désordre*). En effet, la valeur des deux prépositions et les propriétés distributionnelles des arguments N0, N1 et N2 sont codées dans la table[1]. Parmi ces propriétés, on a au moins soit une complétive à l'indicatif (Qu Pind), soit au subjonctif (Qu Psubj), soit, pour l'argument N2, une infinitive coréférente avec l'argument 1 (V1-inf W). Ceci est codé par les colonnes N1 =: Qu Pind, N1 =: Qu Psubj, N2 =: Qu Pind, N2 =: Qu Psubj et N2 =: V1-inf W.

Nous expliquons les modifications de certaines constructions de base (6.1.1), l'éclatement de certaines classes (6.1.2), de simples changements de notation (6.1.3), ou encore la suppression de certaines classes (6.1.4).

6.1.1 Modifications de la construction de base

La construction de base de la table 1 décrite dans Gross (1975) (p. 160) est N0 U Prép V0-inf W (*Paul cesse de travailler*). Tout d'abord, nous pouvons remarquer que la préposition peut être vide. En effet, la colonne Prép1 contenait des −, aujourd'hui remplacés par des <*E*>. Une proposition avait été d'avoir comme construction de base N0 U (E+Prép) V0-inf W, mais cela introduit un *OU* puisque certaines entrées ont les deux formes (*Luc dévale la pente retrouver ses amis / Luc dévale le long du torrent retrouver ses amis*) mais ceci n'est pas une généralité.

Le signe + est une notation que l'on cherche à éviter pour ne pas rendre les intitulés ambigus car il peut avoir trois interprétations différentes :

- toutes les formes sont observées (*ET*) ;
- au moins une des formes est observée (*OU*) ;
- une et une seule des formes est observée (*OU exclusif*).

Comme cela a été expliqué dans Gross (1975), cette table étudie la possibilité d'insérer un verbe U dans une phrase quelconque N0 V W, c'est-à-dire apparaissant dans une phrase de forme N0 U Prép V0-inf W, tout en observant les relations entre N0 et V W,

1. Rappelons que les arguments pouvant être des complétives ne font pas obligatoirement l'objet d'une propriété définitoire, mais parfois de plusieurs colonnes codées dans la table elle-même, ceci pour permettre de distinguer les complétives à l'indicatif, au subjonctif, les complétives interrogatives (*si* P *ou si* P) et les infinitives (V-inf W) et pour permettre également de reconnaître d'autres catégories grammaticales (comme les syntagmes nominaux, avec les propriétés Nhum et N-hum).

pour savoir si le verbe U impose ou non des contraintes sur N0, ce qui est codé par la colonne N0 =: Nnc (non contraint).

Considérons par exemple la phrase :
(Pierre+Cette affaire+Cette table+Faire ceci) amuse Paul
Si l'on y insère le verbe *continuer à -inf*, nous obtenons la séquence tout à fait acceptable, où la relation entre N0 et V W reste inchangée :
(Pierre+Cette affaire+Cette table+Faire ceci) continue à amuser Paul
En revanche, le verbe *se dépêcher de -inf* n'est insérable que dans les phrases à sujet humain :
**(Cette affaire+Cette table+Faire ceci) se dépêche d'amuser Paul*
Pierre se dépêche d'amuser Paul

Or, seul un petit nombre de verbes correspondent à cette étude, à savoir uniquement ceux de la table 1. Le lien entre les deux phrases N0 V W et N0 U Prép V0-inf W n'étant pas exploitable informatiquement et pour rester conforme à la notation des entrées verbales des autres tables, nous avons remplacé U par V. La construction de base de la table 1 est donc N0 V Prép V0-inf W (cf. Fig. 3.3).

La construction de base de la table 3 décrite dans Gross (1975) (p. 168) est N0 V N1 V1-inf W (*On a emmené Luc se faire soigner*). Il s'avère que l'on peut toujours insérer un complément locatif (Loc N2) avant le verbe à l'infinitif, la préposition locative étant codée dans la table (*à*, *dans* ou *de*). Nous avons donc retenu la construction de base la plus complète, qui est N0 V N1 Loc N2 V1-inf W, ce qui n'empêche pas de garder également N0 V N1 V1-inf W comme construction définitoire. De plus, l'infinitive peut être supprimée pour toutes les entrées, donc N0 V N1 Loc N2 est acceptée comme troisième construction définitoire. Reprenons l'exemple précédent pour illustrer ces trois constructions :
On a emmené Luc à l'hôpital se faire soigner
On a emmené Luc à l'hôpital
On a emmené Luc se faire soigner

6.1.2 Éclatement en plusieurs classes

Nous avons procédé à l'éclatement en plusieurs classes quand cela s'est révélé nécessaire. C'est le cas de la table 2 dont la construction de base décrite dans Gross (1975) (p. 165) est N0 V (E+N1) V0-inf avec un *OU exclusif* (*Max fonce retrouver ses copains / La foule envahit la place manifester sa joie*). Cela signifie que certaines entrées acceptent un complément N1 direct et d'autres non, sans que cela ne soit codé. Or, une propriété dite définitoire pour une classe est considérée comme étant acceptée pour toutes les entrées, sans exception. Nous avons donc créé une nouvelle table 2T regroupant toutes les entrées transitives (acceptant un N1), en les supprimant de la table 2. Cela a été fait manuellement pour chaque entrée car aucune indication n'était donnée dans la littérature[2]. De plus, après avoir fait ce découpage, il s'avère que toutes les entrées

2. Nous avons réalisé ce travail en collaboration avec Christian Leclère en 2008 (Tolone, 2009).

non transitives acceptent un complément locatif (Loc N1) qui peut même être parfois obligatoire :
Max s'enfourne dans le tunnel chercher Luc
**Max s'enfourne chercher Luc*
La table 2 a donc comme construction de base N0 V Loc N1 V0-inf W mais également N0 V Loc N1 comme deuxième construction définitoire, car l'infinitive peut être supprimée. On a une colonne N0 V V0-inf W qui code la facultativité du complément locatif et qui est très souvent codée + :
Max fonce dans le jardin retrouver ses copains
Max fonce dans le jardin
Max fonce retrouver ses copains
La table 2T a comme construction de base N0 V N1 V0-inf W et comme deuxième construction définitoire N0 V N1 car l'infinitive peut être supprimée. On a cette même colonne N0 V V0-inf W, codée dans la table 2, qui code ici la possibilité d'omission du N1 et qui est très souvent codée − :
La foule envahit la place manifester sa joie
La foule envahit la place
**La foule envahit manifester sa joie*

6.1.3 Changement de notation

La construction de base de la table 4 décrite dans Gross (1975) (p. 170) est Qu P V N1, avec la colonne N0 =: Nnr codée + pour toutes les entrées (*Que Paul vienne amuse Luc*). Comme cela a été expliqué dans Gross (1975), la notation Nnr désigne un substantif *non restreint*, c'est-à-dire un substantif d'une nature sémantique quelconque, une complétive ou une infinitive. Le test employé consiste à placer, dans la position à étudier, les groupes nominaux *Paul* (humain), *ce gâteau* (concret, énumérable, ou non), *la sincérité* (abstrait), *Qu P* et *V-inf W*. Les phrases résultantes doivent toutes être acceptables. Nous avons par exemple :
(Paul+Ce gâteau+L'amour+Que Paul vienne+Faire cela) ennuie Marie
Nous avons donc supprimé la colonne N0 =: Nnr de la table et l'avons définie comme propriété définitoire. Puis, nous avons transformé la construction de base en N0 V N1, avec N0 =: Qu P, afin de garder la propriété définissant la table, malgré le fait que la complétive Qu P est déjà incluse dans Nnr. Nous avons également ajouté la propriété définitoire N1 =: Nhum.

La construction de base de la table 5 décrite dans Gross (1975) (p. 172) est Qu P V Prép N1 (*Que Marie vienne importe à Paul*). Nous l'avons simplement remplacée par N0 V Prép N1, avec la propriété définitoire N0 =: Qu P. Ce choix de notation permet de réutiliser les mêmes intitulés dans un maximum de tables et facilite la récupération des différentes réalisations de chaque argument. De plus, la table 5 a la propriété définitoire [extrap], qui correspond à la construction Il V N0 Prép N1.

Ce même procédé a été appliqué à la table 12 décrite dans Gross (1975) (p. 201), qui a comme construction de base N0 V Qu Psubj, avec P = N1 V1-inf W (P fai-

sant référence à la forme de la complétive de la construction de base) et comme transformation N0 V N1 de V1-inf W (*Paul apprécie que Léa l'ait aidé / Paul apprécie Léa de l'avoir aidé*). Nous avons remplacé la construction de base par N0 V N1, avec la propriété définitoire N1 =: Qu Psubj. La transformation devient Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W) afin de la rendre plus explicite et est définitoire également. La notation i permet de bien différencier le Ni, sujet de la complétive et le N1, premier complément.

La construction de base de la table 6 décrite dans Gross (1975) (p. 177) est N0 V Qu P (*Paul estime que Luc a raison*). Le fait que l'argument N1 puisse être une complétive est représenté par les trois colonnes N1 =: Qu Pind, N1 =: Qu Psubj et N1 =: si P ou si P codées dans la table. Nous avons gardé N0 V N1 comme construction de base puisque le complément N1 n'accepte pas toujours une complétive. De plus, cela permet de clarifier à quel argument font référence les colonnes précédentes.

La même méthode a été reproduite :

– les tables 7, 8, 9, 10, 14 et 15 ont N0 V à ce Qu P (respectivement, N0 V de Qu P, N0 V Qu P à N2, N0 V Qu P Prép N2, N0 V à ce Qu P Prép N2 et N0 V de ce Qu P Prép N2) comme construction de base dans Gross (1975) (p. 184, 187, 190 et 193) :
  Paul consent à ce que Marie fasse ça (table 7)
  Paul s'insurge de ce que Marie ait fait ça (table 8)
  Paul crie à Léa que tout est fini (table 9)
  Ceci comporte pour Flo qu'elle doit partir tôt (table 10)
  Cette règle équivaut pour Paul à ce que Léa soit élue (table 14)
  Paul répond devant le juge de ce que Léa a fauté (table 15)
  Le fait que l'argument N1 puisse être une complétive est représenté par les deux colonnes N1 =: Qu Pind et N1 =: Qu Psubj codées dans les tables, plus la colonne N1 =: V0-inf W pour les tables 7 et 8, et plus la colonne N1 =: si P ou si P pour la table 9. Nous avons changé leur construction de base en N0 V à N1 pour la table 7, N0 V de N1 pour la table 8, N0 V N1 à N2 pour la table 9, N0 V N1 Prép N2 pour la table 10, N0 V à N1 Prép N2 pour la table 14 et N0 V de N1 Prép N2 pour la table 15. De plus, la table 9 a la propriété définitoire N2 = : Nhum.
– les tables 11 et 13 ont N0 V N1 à ce Qu P (respectivement, N0 V N1 de ce Qu P) comme construction de base dans Gross (1975) (p. 197 et 203) :
  Max a dressé Luc à ce qu'il se lève tôt (table 11)
  Paul informe Luc de ce que Léa arrive (table 13)
  Le fait que l'argument N2 puisse être une complétive est représenté par les trois colonnes N2 =: Qu Pind, N2 =: Qu Psubj et N2 =: V1-inf W codées dans les tables, plus N2 =: V0-inf W pour la table 11. Nous avons remplacé leur construction de base par N0 V N1 à N2 pour la table 11 et N0 V N1 de N2 pour la table 13.
– la table 18 décrite dans Gross (1975) (p. 212), a comme construction de base N0 V Prép N1 Prép N2 (E+pour) Qu Psubj (*Ça a servi à Paul de prétexte pour qu'il ne vienne pas*). Le fait que l'argument N3 puisse être une complétive est représenté par les deux colonnes N3 =: Qu Pind, N3 =: Qu Psubj codées dans la table. Il est vrai

que pour la majorité des entrées, une complétive au subjonctif est acceptée (sauf une entrée qui accepte uniquement une complétive à l'indicatif) et que la préposition en position 3 vaut toujours *pour* ou <*E*>, cependant une information non valable pour toutes les entrées n'a pas à figurer dans la construction de base, que nous avons donc corrigée par N0 V Prép N1 Prép N2 Prép N3.

6.1.4 Suppression de classes

Les tables 17 et 19 ont été supprimées, elles avaient il V (E+Prép ce) Qu P (E+Prép N2) (respectivement, Qu P V N1 Prép N2) comme construction de base dans Gross (1975) (p. 211 et 213) :
Il paraît que Max est franc-maçon (table 17)
Que Léa soit partie ôte tout espoir à Luc (table 19)

La table 19 contenait en effet très peu d'entrées, et elles avaient leur place dans d'autres tables de verbes distributionnels de BGL (Guillet et Leclère, 1992) comme 36DT (*apporter*, *ôter*), 36SL (*désaccoupler*), 37M4 (*remplir*) et 36R (*servir*), qui ont N0 V N1 Prép N2 comme construction de base (ou N0 V N1 à N2 pour 36R), ce qui correspond bien à la construction initiale. Notons tout de même que nous avons accepté une perte d'information pour ces entrées, puisque la propriété N0 =: Qu P n'est pas codée dans ces tables[3].

En ce qui concerne les entrées de la table 17, elles sont considérées comme des expressions figées car le sujet *il* est figé. Nous les avons ajoutées principalement dans la table d'expressions figées de Gross (1982) COQ (*il s'agir pour ... de*, *il apparaître à*, *il paraître*, etc) qui a comme construction de base Det0 C0 V Prép1 (Det1 C1+N1) Prép2 N2.

6.2 Les classes des verbes distributionnels de BGL

Les classes des verbes distributionnels de BGL (cf. Tab. 5.1(a)) étaient au nombre de 40 lors de leur création : 15 classes de constructions transitives (Boons *et al.*, 1976a), 9 classes de constructions intransitives (Boons *et al.*, 1976b) (dont une est considérée actuellement comme une classe d'expressions figées) et 16 classes de constructions transitives locatives (Guillet et Leclère, 1992). Nous allons présenter en détail les modifications apportées aux constructions de base, comme nous l'avons déjà fait dans Tolone *et al.* (2010). Nous commençons par voir les changements de notation (6.2.1), puis les classes pour lesquelles nous avons dû spécifier les propriétés distributionnelles de leurs arguments (6.2.2). Ensuite, nous détaillons le cas de deux classes qui ont été divisées chacune en quatre classes (6.2.3), puis des trois classes qui n'ont jamais été publiées (6.2.4), et enfin d'une nouvelle classe qui a été créée (6.2.5). Puis, un exemple d'ajout d'intitulé est donné (6.2.6), avant de terminer par un inventaire des classes restantes, n'ayant pas subi de modification (6.2.7).

3. La propriété N0 =: Qu P devrait être codée dans toutes les tables car elle est variable selon les entrées (codage O). En effet, les tables de M. Gross couvrent bien l'ensemble des complétives objet mais pas des complétives sujet, la table 5 regroupant uniquement celles entrant dans la construction N0 V Prép N1.

6.2.1 Changement de notation

La construction de base de la table 32R1 (Résiduel à restructuration) décrite dans Boons *et al.* (1976a) (p. 76) est N0 V N1 de N1c = N0 V N1c Loc N1 (*Ceci confirme les soupçons de Marie /Ceci confirme Marie (*E+dans ses soupçons)*). Nous avons décidé de noter la construction de base simplement N0 V N1. De plus, nous y ajoutons comme définitoire, la restructuration N0 V (N1 de N1c) = N0 V (N1c) (Prép N1). Elle signifie que l'objet direct nominal de la forme N1 de N1c, où N1c est un complément de nom, peut être remplacé par N1c et le complément prépositionnel Prép N1. Nous avons également ajouté les propriétés définitoires N1 =: Nabs et [passif par]. Le complément prépositionnel peut être facultatif (*Luc stimule la curiosité de Marie / Luc stimule Marie (E+dans sa curiosité)*). Ceci est codé dans la table par la colonne N0 V N1c.

La construction de base de la table 34L0 (Locatif en position N0) décrite dans Boons *et al.* (1976b) (p. 242) est la construction standard Ni V Loc Nj et la construction croisée Nj V de Ni (*L'intelligence étincelle dans son regard / Son regard étincelle d'intelligence*). Tout d'abord, la notation i et j d'origine a été abandonnée : N0 V Loc N1 et N1 V de N0. Puis, nous avons renommé la construction croisée N1 V Prép N0 avec Prép0 =: de définitoire, car dans la table, une deuxième préposition est possible et codée par Prép0 =: en. Il y a également la propriété N1hum V Prép N0 qui est codée dans la table, mais elle ne sert qu'à spécifier que le N1 est humain.

La construction de base de la table 38L (Locatif double) décrite dans Guillet et Leclère (1992) (p. 123 et 219) est N0 V N1 de N Loc N, le complément de N Loc N n'étant pas numéroté et étant défini par deux possibilités :

– de N Loc N est indissociable, comme dans :
  Max balade le spot de la porte à la fenêtre
  **Max balade le spot de la porte*
  **Max balade le spot à la fenêtre*
– le premier N est interprété comme une source *dépendante*, c'est-à-dire employée seulement en présence de la destination, et le deuxième comme une destination, ainsi :
  Max traîne cette valise du salon dans la chambre
  **Max traîne cette valise du salon*
  Max traîne cette valise dans la chambre

La numérotation étant indispensable, surtout pour indiquer la possibilité d'effacement du premier complément, nous avons changé la construction de base en N0 V N1 Loc N2 source Loc N3 destination, avec les deux propriétés définitoires N2 lieu source de N1 et N3 lieu de destination de N1. De plus, contrairement à la définition, les sources n'étant pas toutes dépendantes (*Max transporte du blé de Gap (E+à Dax)*), les deux colonnes N0 V N1 de N2 source et N0 V N1 Loc N3 destination sont codées dans la table. On a de plus le fait que la propriété N1 =: Nhum n'est pas *obligatoire*[4], ce

4. Ceci est à mettre en opposition avec la table 38LH qui a la propriété N1 =: Nhum obligatoire, ce qui implique que la propriété N1 =: N-hum est codée −.

que nous avons traduit par le codage + de la propriété N1 =: N-hum, la propriété N1 =: Nhum restant à coder (codage O).

La construction de base de la table 36SL (Symétrique Locatif) décrite dans Guillet et Leclère (1992) (p. 229) est la construction locative standard N0 V N1 (avec+Loc) N2, avec la possibilité d'interchanger N1 et N2 (qui n'est pas codée) et la construction *symétrique* N0 V N1 et N2. Elle reconnaît :
Luc colle la fiche bleue sur la fiche jaune
Luc colle la fiche jaune sur la fiche bleue
Luc colle la fiche bleue et la fiche jaune
L'objet N2 est interprété comme un lieu, soit N2 lieu source de N1, soit N2 lieu de destination de N1, mais ces deux colonnes sont à coder (codage O). La colonne Prép2 =: avec est codée dans la table. Or, quand cette propriété n'est pas acceptée, c'est la propriété Prép2 =: d'avec qui est vraie, nous l'avons donc ajoutée pour la coder explicitement avec le codage inverse. Nous avons défini N0 V N1 Prép N2 comme construction de base, qui est à associer aux deux colonnes Prép2 =: avec et Prép2 =: d'avec ; et N0 V N1 Loc N2 comme deuxième construction définitoire, qui est à associer aux différentes prépositions locatives pour le complément N2 codées dans la table (*de*, *dans*, *sur*, *contre* et *à*) ; sans oublier la construction symétrique N0 V N1 et N2 également comme construction définitoire (*Max a cogné son verre (avec+contre+et) celui de Luc*).

La construction de base de la table 38LD (Locatif Destination) décrite dans Guillet et Leclère (1992) (p. 123 et 153) est N0 V N1 Loc N2, avec N2 destination, que nous avons remplacée par la construction de base N0 V N1 Loc N2 destination (*Max applique du vernis sur le meuble*). Nous avons ajouté les propriétés définitoires avec N2 lieu de destination de N1 et N1 =: N-hum.

La construction de base de la table 38LS (Locatif Source) décrite dans Guillet et Leclère (1992) (p. 123 et 167) est N0 V N1 Loc N2, avec N2 source, que nous avons codée N0 V N1 Loc N2 source (*Max a arraché le clou de la planche*). Nous avons ajouté les propriétés définitoires avec N2 lieu source de N1 et N1 =: N-hum.

La construction de base de la table 38LR (Locatif Résiduel) décrite dans Guillet et Leclère (1992) (p. 123 et 175) est N0 V N1 Loc N2 (sans source ni destination : *Paul a agencé les bibelots sur la cheminée*). Nous avons ajouté les propriétés définitoires N2 lieu statique de N1 et N1 =: N-hum.

6.2.2 Spécification de propriétés distributionnelles

Pour un grand nombre de classes, la construction de base est bien représentée mais il manque des propriétés distributionnelles spécifiques pour un argument, contenant pour la plupart des informations sémantiques.

La construction de base de la table 32A (Apparition) décrite dans Boons *et al.* (1976a)

(p. 12) est N0 V N1 apparition, ce qui signifie que l'objet N1 est interprété comme apparaissant ou étant créé au cours du procès. C'est le cas dans *Max a bâti une maison*, où la maison n'existe pas avant que l'action ne se soit déroulée. Dans un souci de réutilisabilité des intitulés et de simplification des informations sémantiques des arguments, nous l'avons séparée en deux : la construction de base N0 V N1 et la propriété définitoire N1 apparition. Nous avons ajouté la propriété définitoire [passif par]. De plus, il est spécifié que la moitié des entrées acceptent un complément Prép N2 qui peut être de deux types :

– (dans+de+en) N2, codé par les propriétés N0 V N1 de N2 matériau et N0 V N1 Loc N2 matériau, Loc # de[5], marquant le matériau qui a servi à constituer le N1 (*Jean a construit une maison (de+en) briques creuses*) ;
– Loc N2, codé par la propriété N0 V N1 Loc N2, qui marque explicitement un rapport de lieu (interprété comme une destination) entre N1 et N2 (*Jean a construit sa maison sur un terrain argileux*).

Ces compléments sont quelquefois compatibles à l'intérieur d'une même phrase (*Jean a construit une maison en brique dans son champ*), ce qui pose problème puisque les deux sont numérotés N2. La table devrait sans doute être dédoublée comme cela a été fait pour les tables 35L et 38LH (cf. 6.2.3).

La construction de base de la table 32CV (ConVersion) décrite dans Boons *et al.* (1976a) (p. 14) est N0 V N1 (E+en V-n), V-n désignant le nom issu du verbe V (*Max a roulé les papiers (E+en un mince rouleau)*). Afin d'éviter le +, ayant pour valeur *ET* ici, qui peut prêter à confusion, nous avons choisi de mettre deux constructions définitoires (mais pas de base), qui sont N0 V N1 et N0 V N1 en V-n. De plus, il est spécifié que tous les verbes contiennent un substantif qui dénote le résultat du procès et qui est interprété comme une apparition après un processus de conversion (*caraméliser du sucre = le (convertir+transformer) en caramel*, *torsader des fils = les mettre en torsade*). C'est pourquoi nous avons également défini la propriété définitoire N2 apparition et N2 être V-n, le reste étant codé par les deux colonnes V = convertir en V-n et V = mettre en V-n. Enfin, nous avons mis N0 V N1 en N2 comme construction de base (et non pas N0 V N1 en V-n) car le V-n est difficile à prendre en compte dans les systèmes de TAL et peut se généraliser ici par un syntagme nominal quelconque (*Max a roulé les papiers en boule*). Cela permet également de clarifier la position du N2 pour la propriété définitoire N2 apparition.

La construction de base de la table 38PL (N2 Pluriel) décrite dans Boons *et al.* (1976a) (p. 5 et 19) est N0 V N1hum en N2 plur obl, ou plutôt N0 V N1-hum en N2 plur obl après rectification (*Max analyse la phrase en deux constituants*). Pour les mêmes raisons que précédemment, les différentes informations sémantiques des arguments étant codées à part, nous avons remplacé la construction de base par N0 V N1 en N2 avec N2 =: Npl obl. De plus, nous avons supprimé la colonne N1 =: N-hum de la table afin de la rendre définitoire également.

5. Loc # de désigne une préposition locative différente de *de*.

La construction de base de la table 32PL (Pluriel) décrite dans Boons *et al.* (1976a) (p. 5 et 24) est N0 V N1 plur obl (E+en N2) (*Max compile des données (E+en tables)*). La colonne N0 V N1 en N2 étant codée dans la table, nous avons gardé comme construction de base N0 V N1, avec la propriété définitoire N1 =: Npl obl. Nous avons ajouté la propriété définitoire [passif par]. D'autres colonnes sont codées dans la table : N0 V N1 de N2 source, N0 V N1 Loc N2 source, Loc # de, N0 V N1 Loc N2 source Loc N3 destination, ainsi que différentes prépositions locatives pour le complément N3 telles que *dans*, *sur*, *contre* et *à*. La propriété N0 V N1 Loc N3 destination devrait peut-être également être ajoutée.

La construction de base de la table 32CL (Corps Locatif) décrite dans Boons *et al.* (1976a) (p. 66) est N0 V N1pc de N1c, N1c étant un complément du nom N1pc (*Jean a attrapé le bras de Paul*). Nous avons simplifié la notation du complément N1pc de N1c par N1 en admettant comme construction de base N0 V N1, avec N1 =: Npc. Nous avons également ajouté les propriétés définitoires N0 V N1 Loc N1pc W et [passif par].

La construction de base de la table 32C (Concret) décrite dans Boons *et al.* (1976a) (p. 69) est la même que pour la table 32CL, sa construction de base est N0 V N1 (*L'immobilité a ankylosé le genou de Paul*), mais la propriété N1 =: Npc est variable. Nous avons également ajouté les propriétés définitoires N1 = : N-hum et [passif par]. Enfin, la table 32C a la propriété distributionnelle N1 =: Nconc définitoire (p. 73).

Il est important de remarquer que tous les Npc sont des Nconc, mais l'inverse n'est évidemment pas vrai. Or la définition de la table 32C a évolué, elle accepte des Nconc qui ne sont pas des Npc (*Luc a abîmé le livre*). C'est pourquoi la colonne N1 =: Npc n'est pas définitoire mais variable, il faudrait donc la coder (codage O). La construction de base de la table 32CL est donc N0 V N1, avec N1 =: Npc, alors que celle de la table 32C est N0 V N1, avec N1 =: Nconc.

Les propriétés suivantes sont décrites également comme étant définitoires mais sont en fait codées dans les tables : pour la table 32CL, N0 lui V N1pc W et N0 V N1c Loc N1pc W ou plutôt N0 V N1 Loc N1pc W car la distinction entre N1 et N1c n'est pas faite ici ; et pour la table 32C, N0 lui V N1pc W et N1 être Vpp W.

La table 35S (Symétrique), décrite dans Boons *et al.* (1976b) (p. 207), regroupe les verbes acceptant la construction standard N0 V Prép N1, avec Prép =: avec+d'avec et la construction symétrique N0 et N1 V (*Max boxe avec Luc / Max et Luc boxent (E+ensemble+l'un avec l'autre)*). Or, les colonnes Prép =: avec, Prép =: d'avec et d'autres (*dans*, *à* et *de*) sont codées dans la table, mais Prép =: avec étant codée + pour toutes les entrées, nous avons pu la supprimer de la table. En revanche, Prép =: d'avec doit figurer dans la table puisqu'elle n'est acceptée que pour certaines entrées. Nous avons donc gardé N0 V Prép N1 comme construction de base, avec Prép =: avec définitoire, et N0 et N1 V comme deuxième construction définitoire.

La construction de base de la table 36DT (DaTif) décrite dans Guillet et Leclère (1992) (p. 123 et 237) est N0 V N1 à N2, avec N0 = : Nhum et N2 = : Nhum (il y a une notion d'échange entre N0 et N2, qui sont tous les deux humains). Le complément N2 est

soit interprété comme bénéficiaire du référent du N1 (*Paul accorde un prêt*), soit comme perdant le référent du N1 (*Luc a acheté cet immeuble à Marie*, où Marie ne possède plus l'immeuble après le procès). Le premier cas est codé par la colonne N2 bénéficiaire. Nous avons explicité le deuxième cas en ajoutant la colonne N2 détrimentaire, contenant le codage inverse. Le complément N2 peut se pronominaliser en Ppv =: lui (*lui* est pris ici comme représentant canonique de la classe des Ppv au datif : *me*, *te*, *lui*, *se*, *nous*, *vous*, *leur*). La table reconnaît par exemple *Max passe un stylo à Ida* et *Max lui passe un stylo*. La préposition pouvant être *à* ou *de* (*Max a perçu son salaire de Luc*), deux colonnes sont codées dans la table : Prép2 =: à et N0 V N1 de N2 source (notation qui permet d'indiquer également que le complément N2 est interprété comme une source). Les propriétés distributionnelles N0 = : Nhum et N2 =: Nhum sont définitoires. En revanche, la propriété distributionnelle N2 =: N-hum étant codée dans la table, nous avons dupliqué le codage de la colonne pour la propriété distributionnelle N0 = : N-hum qui n'était pas codée. De plus, Ppv =: lui désigne ici la propriété définitoire mais également la propriété codée dans la table dépendante de N2 =: N-hum. Nous avons renommé la propriété définitoire Prép N2hum = Ppv =: lui. Quant à la colonne nommée Ppv =: lui, nous l'avons renommée Prép N2-hum = Ppv =: lui (*Ceci retire du charme à la maison* et *Ceci lui retire du charme*). De plus, comme le N2 (qu'il soit humain ou non) peut toujours se pronominaliser en *lui*, elle contient également le même codage que N2 =: N-hum, soit un − lorsqu'il n'y a pas de N2-hum.

La construction de base de la table 32RA (Rendre Adj) décrite dans Boons *et al.* (1976a) (p. 6 et 34) est N0nr V N1 (E+de N2) (*La rouille affaiblit cette poutre*). La construction de base est donc N0 V N1, avec N0 =: Nnr définitoire (*(Jean+La chaleur+Le produit solaire+Rester sur la plage) a séché l'épiderme de Marie*). Nous avons ajouté la propriété définitoire [passif par]. Cette table regroupe des verbes en relation morphologique avec un adjectif et accepte également la construction N0 rendre N1 (E+plus) V-adj (*La rouille rend (E+plus) faible cette poutre*). Le + dans cette construction a une valeur de *OU exclusif*, puisque certaines entrées acceptent uniquement la propriété N0 rendre N1 V-adj (sans adverbe comparatif), comme *Ce produit active la fermentation = Ce produit a rendu la fermentation active*, et d'autres uniquement la propriété N0 rendre N1 plus V-adj (avec adverbe comparatif), comme *Le vent a abaissé la température = Le vent a rendu la température plus basse*. La construction N0 rendre N1 (E+plus) V-adj a été dédoublée et codée dans la table[6]. De plus, certains verbes acceptent un complément prépositionnel de N2, mais pas tous. Le + dans N0nr V N1 (E+de N2) a donc une valeur de *OU exclusif*, c'est pourquoi le complément de N2 ne figure pas dans la construction de base. Ce complément peut être de trois types comme décrit p. 51 :

– N0 V N1 de combien : *Jean a allongé son chemin de 30 mètres*
  Ce complément supplémentaire, introduit par la préposition *de*, précise une quantité

6. Grâce au codage de certaines informations concernant les V-n, et en l'occurrence les V-adj, dans les tables de verbes (Paumier, 2003). Ce codage n'ayant pas été maintenu dans la version actuelle des tables, il a été réintégré par Sébastien Paumier début 2011.

ou une mesure intéressant le procès, est souvent à déterminant numéral et répond à la question en *de combien ?* ;
– N0 V N1 de N2 (enlever) : *Luc a purgé ce radiateur d'un peu d'air* qui peut être paraphrasée par N0 enlever N2 de N1 (*Luc a enlevé un peu d'air de ce radiateur*) ;
– N0 V N1 de N2 (mettre) : *Paul a alourdi son sac d'un gros livre* qui peut être paraphrasée par N0 mettre N2 Loc N1 (*Pierre a mis un gros livre dans son sac*).

Dans la table, trois colonnes sont codées :
de combien ?
N0 V N1 de N2 = N0 enlever N2 Loc N1 (à l'origine V = enlever)
N0 V N1 de N2 = N0 mettre N2 Loc N1 (à l'origine V = mettre)
Ces deux dernières codant uniquement les paraphrases, nous avons ajouté la colonne N0 V N1 de N2, valant + dès qu'une des deux colonnes précédentes vaut +. Cette information est redondante mais permet d'expliciter la construction non déductible autrement. De plus, cela rend la notation = conforme à son utilisation, qui est de spécifier une propriété transformationnelle (à droite) à partir d'une construction déjà existante (à gauche).

La construction de base des tables 37E (Enlever) et 37M1 à 37M6 (Mettre) décrite dans Guillet et Leclère (1992) (p. 123 et 130) est N0 V N1 de N2, avec les propriétés N1 source pour la table 37E, et N1 destination pour les tables 37M1 à 37M6. Nous n'avons pas gardé la propriété N1 source pour la table 37E car en réalité cette table accepte soit la propriété N1 lieu source de N2 (*Max cure le puits de sa vase*), soit la propriété N1 détrimentaire (*Max a possédé Luc de 100 euros*), colonnes qui sont à coder (codage O). De même, pour la table 37M1 (non locatif), la destination est sémantique et non locative, elle a le sens de bénéficiaire (*Max goinfre Bob de gâteaux*). Nous avons considéré qu'un argument bénéficiaire ne peut pas être en même temps un lieu, ce n'est donc pas la propriété N1 destination qui est définitoire mais la propriété N1 bénéficiaire. Cette propriété distingue la table 37M1 des tables 37M2 à 37M6, de même qu'elle distingue la table 36DT des tables 38LD à 38LHR (voir l'annexe G). Pour les tables 37M2 à 37M6, la propriété définitoire a été renommée N1 lieu de destination de N2 (*Luc sale le rôti de sel fin*).

La seule préposition possible pour la table 37E étant *de*, la construction de base est N0 V N1 de N2. En revanche, la construction de base des tables 37M1 à 37M6 est N0 V N1 Prép N2 avec Prép2 =: de, pour permettre de coder dans les tables les prépositions supplémentaires *avec*, *dans* et *en*.

Le sens des verbes en 37E est *enlever* (*Max débarrasse le lit des oreillers = Max enlève les oreillers du lit*), la paraphrase N0 V N1 de N2 = N0 enlever N2 de N1 est donc définitoire. Celui des verbes en 37M1 à 37M6 est *mettre* (*Max couvre le lit d'oreillers = Max met des oreillers sur le lit*), codé par la paraphrase N0 V N1 de N2 = N0 mettre N2 Loc N1 définitoire.

Dans les tables 37M1 à 37M6, la productivité du préfixe *dé-* permet de fabriquer un grand nombre de verbes de type 37E. Ces verbes n'apparaissent pas en tant qu'entrées du lexique, mais sont à reconstruire lorsque la colonnes dé-V est dans l'une des tables 37M1 à 37M6. Cette colonne n'est pas exploitée car le préfixe dépend des verbes (*dé-* pour *coloniser*, à contraster avec *dés-* pour *occuper*).

La différence entre les tables 37M2 à 37M6 est principalement sémantique ou morphologique. De plus, les critères justifiant leur découpage ne sont pas définissables clairement et n'ont pas la moindre reproductibilité, c'est pourquoi ces tables ont été regroupées dans l'annexe F (voir section 7.3 avec N2 =: déformation). Cela a tout de même été codé pour justifier l'existence de chacune des tables et ne rien laisser implicite :

- dans la table 37M2 (trace), le corrélat N2 du lieu N1 est un substantif indiquant des traces ou des marques (*Paul a paginé ses feuilles de numéros fantaisistes*). La propriété N2 =: trace a été ajoutée et est définitoire pour cette table ;
- dans la table 37M3 (déformation), on a une modification du lieu N1 : le N2 ne dénote pas un objet concret adjoint à N1, mais le type de déformation subit par ce dernier (*Paul a cabossé le plateau de grosses bosses*). La propriété N2 =: déformation a été ajoutée et est définitoire pour cette table ;
- dans la table 37M4 (résiduel), le N1 est clairement un lieu, et le N2, corrélat de ce lieu, représente un objet concret non relié morphologiquement au verbe (*Paul parsème la table de miettes*) ;
- dans les tables 37M5 et 37M6 (apport), le N2, corrélat du lieu N1, est de la forme N de V-n (respectivement, V-n de N), le V-n étant le représentant canonique de la classe des substantifs possibles dans cette position :
  On a empierré la route de (E+une couche de) pierres (table 37M5)
  Paul assaisonne son plat de (E+un assaisonnement de) épices exotiques (table 37M6)
  La propriété N2 =: N de V-n a été ajoutée et est définitoire pour la table 37M5, ainsi que N2 =: V-n de N pour la table 37M6.

De plus, les tables 37M1 à 37M6 avaient comme propriétés distributionnelles codées dans les tables N1 =: Nhum obl et N2 =: Nhum obl. Lorsque par exemple, N1 =: Nhum obl est codée +, cela signifie que N1 =: Nhum est vraie et N1 =: N-hum est fausse. Mais lorsque N1 =: Nhum obl est codée −, cela signifie que soit les deux propriétés précédentes sont vraies, soit uniquement N1 =: N-hum est vraie. La notation hum obl a donc été supprimée puisqu'elle ne permet pas de renseigner complètement la propriété N1 =: Nhum.

Nous avons donc dupliqué la colonne N1 =: Nhum obl (respectivement, N2 =: Nhum obl) en la renommant d'une part en N1 =: Nhum (respectivement, N2 =: Nhum) en substituant tous les − par des ∼, et d'autre part en N1 =: N-hum (respectivement, N2 =: N-hum) en inversant les + et les −. Bien sûr, le codage reste à compléter pour les colonnes N1 =: Nhum et N2 =: Nhum, mais cela à l'avantage de coder explicitement les deux colonnes, comme cela est fait dans toutes les autres tables.

La construction de base de la table 32R2 (Résiduel) décrite dans Boons *et al.* (1976a) (p. 79) est N0 V N1 sans aucune autre contrainte (*Les pluies ont accentué les crues*). Nous avons ajouté les propriétés définitoires N1 =: Nabs et [passif par].

6.2.3 Éclatement en plusieurs classes

Comme nous l'avons mentionné en 6.1.2, nous avons dédoublé la table 2 car elle acceptait un complément N1 direct pour certaines entrées et pour d'autres non. La nou-

velle table 2T regroupe donc toutes les entrées transitives. Pour des raisons totalement différentes, nous avons également divisé la table 35L, ainsi que la table 38LH, comme nous le montrons à présent [7].

La construction de base des tables 35L (Locatif) et 35ST (STatique) décrite dans Boons *et al.* (1976b) (p. 216 et 235) est N0 V Loc N1. Pour la table 35L, le N1 est interprété comme un lieu source et/ou destination (*Le bateau s'enfonce dans les flots*), alors que dans la table 35ST sont regroupés les emplois *statiques* (*Le pieu sort de l'eau*). Le complément Loc N1 de la table 35L correspond donc à un complément interprété comme une source, que l'on note N1, ou un complément interprété comme une destination, noté N2, ou les deux en même temps. Cela pose problème pour savoir à quel complément font référence les propriétés codées dans la table, puisque la construction de base n'en contient qu'un seul. C'est pourquoi nous avons divisé la table 35L en cinq tables :

- 35L (Locatif) avec comme construction de base N0 V Loc N1 source Loc N2 destination, reconnaissant :
  Paul a bondi du tabouret sur la table
  Paul a bondi du tabouret
  Paul a bondi sur la table
  Ici, les deux compléments sont acceptés ensemble ou chacun séparément. Cela est codé dans la table par les deux constructions N0 V Loc N1 source et N0 V Loc N2 destination. Ce qui permet de déterminer les sources dépendantes, où seule la destination peut apparaître isolément, comme dans :
  Max chemine de chez lui vers Gap
  **Max chemine de chez lui*
  Max chemine vers Gap ; De plus, nous avons ajouté les deux propriétés définitoires N1 lieu source de N0 et N2 lieu de destination de N0.
- 35LS (Locatif Source) avec comme construction de base N0 V Loc N1 source, c'est-à-dire ne reconnaissant pas de destination (*Le train a déraillé de la voie*). De plus, la préposition *de* étant toujours possible, la propriété Loc N1 =: de N1 source est définitoire. Nous avons ajouté la propriété définitoire N1 lieu de destination de N0 ;
- 35LD (Locatif Destination) avec comme construction de base N0 V Loc N1 destination, c'est-à-dire ne reconnaissant pas de source (*Le bateau s'enfonce dans les flots*). Nous avons ajouté la propriété définitoire N1 lieu source de N0 ;
- 35ST (locatif STatique) avec comme construction de base N0 V Loc N1, N1 étant le lieu statique de N0. Cette table rassemble les emplois statiques des verbes de mouvement (*Le pieu sort de l'eau*, où le pieu ne bouge pas) ou des verbes sans mouvement (*Max habite à Paris*). Nous avons ajouté la propriété définitoire N1 lieu statique de N0 ;
- 35LR (Locatif Résiduel) avec comme construction de base N0 V Loc N1. Cette table concerne les mouvements internes à un lieu sans déplacement (*Max appuie sur le bouton*). Nous avons ajouté la propriété définitoire N1 = où, signifiant que l'objet N1 répond à la question en *où ?* sans être le lieu de N0 (*Où le livre est-il paru ? - Il*

7. Nous avons réalisé ce travail en collaboration avec Christian Leclère en 2010 (Tolone *et al.*, 2010).

est paru chez Plon). Le fait de distinguer les locatifs résiduels a fait changer de table la moitié des entrées de la table 35ST.

La construction de base de la table 38LH (Locatif à corrélat Humain) décrite dans Guillet et Leclère (1992) (p. 123 et 202) est N0 V N1 Loc N2, avec N1 =: Nhum obligatoire (et donc N1 =: N-hum codée −), et dont le N2 est interprété comme un lieu source et/ou destination (*On a viré Max de son poste*). Les prépositions locatives sont codées dans la table : *de* ou une autre préposition (Loc # de) pour l'argument source et *dans*, *sur*, *contre*, *à* ou *vers* pour l'argument destination. De même que pour la table 35L, si l'on note N2 le complément interprété comme une source et N3 celui interprété comme une destination, l'un, l'autre ou les deux peuvent apparaître, ce qui est contradictoire avec la construction de base. Nous avons donc divisé la table 38LH en quatre tables :

- 38LH (Locatif à corrélat Humain) avec comme construction de base N0 V N1 Loc N2 source Loc N3 destination, avec N1 =: Nhum obligatoire, qui accepte les deux compléments ensemble (*Le général a replié ses soldats du champs de bataille sur leurs lignes*). De plus, nous avons ajouté les deux propriétés définitoires N2 lieu source de N1 et N3 lieu de destination de N1. La table code également les deux constructions N0 V N1 Loc N2 source et N0 V N1 Loc N3 destination, permettant d'accepter chacun des compléments séparément :
  Le général a replié ses soldats du champs de bataille
  Le général a replié ses soldats sur leurs lignes
  Cela permet de déterminer les sources dépendantes, comme dans :
  Max conduit Ida de la chambre au salon
  **Max conduit Ida de la chambre*
  Max conduit Ida au salon ;
- 38LHS (Locatif Source à corrélat Humain) avec comme construction de base N0 V N1 Loc N2 source, avec N1 =: Nhum obligatoire, et N2 lieu source de N1 (sans destination : *On a viré Max de son poste*) ;
- 38LHD (Locatif Destination à corrélat Humain) avec comme construction de base N0 V N1 Loc N2 destination, avec N1 =: Nhum obligatoire, et N2 lieu de destination de N1 (sans source : *Max a engagé son fils dans la mairie*) ;
- 38LHR (Locatif Résiduel à corrélat Humain) avec comme construction de base N0 V N1 Loc N1, avec N1 =: Nhum obligatoire, et N2 lieu statique de N1, concernant les mouvements internes à un lieu (*Max sème Ida dans le métro*).

6.2.4 Classes non publiées

Même si ces classes n'ont jamais été publiées, leur construction de base est décrite dans l'inventaire de Leclère (1990) et sur le site `http://infolingu.univ-mlv.fr/` (Données Linguistiques > Lexique-Grammaire > Visualisation).

La construction de base de la table 36S (Symétrique), décrite également dans Borillo (1971), est N0 V N1 (avec+à) N2 ou N0 V N1 (d'avec+de) N2, mais aussi N0 V N1 et N2 en relation de paraphrase (*Le maire a marié Paul (avec+à+et) Marie / On a dissocié Paul (d'avec+de+et) Luc*). Les colonnes Prép2 =: à, Prép2 =: de, Prép2 =: avec et

Prép2 =: d'avec étant codées dans la table, nous avons noté la construction de base simplement N0 V N1 Prép N2 et la deuxième construction définitoire N0 V N1 et N2.

La construction de base de la table 35RR (Résiduel double) est N0 V Prép N1 Prép N2 (*Paul rivalise d'astuce avec Jean*). Cette table n'a été décrite dans aucun livre et ses intitulés sont en cours d'élaboration. Notons que la préposition est toujours différente de *à* et de la préposition vide (<E>), la propriété Prép2 =: à est donc codée −.

La construction de base de la table 38RR (Résiduel double) est N0 V N1 Prép N2 Prép N3 (*Paul offre de l'argent à Luc pour ce travail*). Cette table est résiduelle par rapport à la table 38L, car elle n'accepte pas les deux propriétés N2 lieu source de N1 et N3 lieu de destination de N1 en même temps. Les deux prépositions sont codées dans la table (*à*, *en*, *de*, *avec*, *contre*, *pour* ou *auprès de* pour la préposition 2 et *de*, *à*, *avec*, *par*, *pour*, *contre*, *comme*, *dans*, *en* ou *sur* pour la préposition 3), même si la plupart ne pas encore codées (codage ∼).

6.2.5 Ajout de classe

La table 32D (Disparition) a été créée parallèlement à la table 32A (cf. 6.2.2) même si elle comporte peu d'entrées. L'objet N1 est interprété comme disparaissant au cours du procès (*Max a démoli la maison*). Sa construction de base est N0 V N1, avec N1 disparition. Nous avons ajouté la propriété définitoire [passif par]. La liste des entrées codées dans cette classe est actuellement : *anéantir*, *démolir*, *détruire*, *fusiller*, *sacrifier*, *souffler*, *supprimer*, *volatiliser*.

6.2.6 Ajout d'intitulé

La construction de base de la table 38L0 (Locatif en position N0) décrite dans Guillet et Leclère (1992) (p. 123 et 193) est N0 V N1 (*Max conserve son chapeau (E+sur la tête)*). Nous avons ajouté la propriété définitoire N1 =: N-hum. Le sujet est un lieu, ce qui est codé dans la table par les colonnes N0 lieu source de N1 et N0 lieu de destination de N1, lorsque le verbe a un sens de mouvement, et n'est pas codé dans le cas où le verbe a un sens statique (propriété N0 lieu statique de N1 qui reste à coder). Nous avons ajouté la colonne N0 V N1 Loc N2 dans la table pour coder la possibilité de rajouter un complément locatif.

6.2.7 Reste des classes

Les classes suivantes n'ont pas subi de modification dans leur construction de base, mais nous l'avons ajoutée à la table des classes.

Nous décrivons les classes de constructions transitives qui n'ont pas été décrites précédemment et dont les pages citées correspondent à la description de la construction de base dans Boons *et al.* (1976a), les classes de constructions intransitives décrites dans Boons *et al.* (1976b), ainsi que les classes de constructions transitives locatives décrites dans Guillet et Leclère (1992).

Classes de constructions transitives

La construction de base de la table 36R (Résiduel) est N0 V N1 à N2 (p. 58) (*Max attache une signification à ce geste*). Cette table est résiduelle car les compléments à N2 ne correspondent pas à ceux qui permettent de définir de grandes classes d'emplois (comme les datifs, symétriques ou locatifs).

La construction de base de la table 38R (Résiduel) est N0 V N1 Prép N2 (p. 9 et 60) (*Jean commence son discours par une citation*), où la Prép2 est variable, elle est donc codée dans la table. Notons que la préposition est toujours différente de *à*, la propriété Prép2 =: à est donc codée −.

La construction de base de la table 39 est N0 V N1 N2 (p. 62) (*On a élue Marie présidente*).

La construction de base de la table 32NM (NuMérique) est N0 V N1, avec passif interdit (p. 73), les deux propriétés [passif par] et [passif de] sont donc codées − (*Ce livre comprend dix chapitres*).

La construction de base de la table 32H (Humain) est N0 V N1, avec N1 =: Nhum obligatoire (p. 75) (*Le bruit a abasourdi Max*).

La construction de base de la table 32R3 (Résiduel semi-figé) est N0 V N1 (p. 81) (*Luc occupe (un poste+une charge+un emploi)*). Nous avons ajouté la propriété définitoire [passif par]. Cette table code une colonne intitulée thèmeN1, contenant le *classifieur* de la petite classe restreinte des noms qui peuvent occuper la position objet direct pour l'emploi concerné (par exemple, ici *fonction*). L'unité sémantique et syntaxique semble être le couple V N1 plutôt que V lui-même, et une solution serait de classer ces emploi dans les constructions à verbe support.

Classes de constructions intransitives

La construction de base de la table 35R (Résiduel) est N0 V Prép N1 (p. 253) (*Max a collaboré avec l'occupant*). Un grand nombre de prépositions sont possibles, elles sont codées dans la colonne Prép1. Il s'agit d'une table résiduelle qui n'a pas de propriété définitoire comme les autres tables (symétriques, locatifs, complétives). Elle contient une colonne intitulée thèmeN1 qui restreint le N1 (par exemple, ici *ennemi*).

La construction de base de la table 33 est N0 V à N1 (complément non locatif) (p. 252) (*Le caporal s'est rendu à l'ennemi*, cf. Fig. 3.1). Il s'agit d'une table résiduelle par rapport aux tables 1, 5 et 7 admettant des complétives (sujet pour la table 5, en N1 pour la table 7) ou des infinitives (en N1 pour la table 1).

La construction de base de la table 31R (Résiduel) est N0 V, avec N0 =: N-hum (*La route dérape*), puisque si N0 =: Nhum est obligatoire, l'entrée figure dans la table 31H (p. 262). C'est une table résiduelle car les compléments ne sont pas homogènes, dont le sujet, qui n'est pas systématiquement contraint.

La construction de base de la table 31H (sujet Humain) est N0 V, avec N0 =: Nhum (p. 259) (*Max déambule (E+dans la maison)*). La propriété distributionnelle N0 =: Nhum est obligatoire, même si la colonne N0 =: N-hum métaphore est codée dans la table pour permettre de reconnaître une phrase métaphorique (*Son imagination déambulait li-*

brement). Certains compléments non essentiels, mais caractéristiques pour certains verbes, ont été codés grâce aux colonnes N0 V vers N (*Paul titube (E+vers la porte)*) et N0hum V W sur ce point (*Paul se goure (E+sur ce point)*).

Classes de constructions transitives locatives

La construction de base de la table 38L1 (Locatif en position N1) est N0 V N1 (p. 123 et 214) (*L'avion survole le pays*). Nous avons ajouté la propriété définitoire N1 =: N-hum. L'objet N1 est interprété comme un lieu, ce qui est codé dans la table par les colonnes N1 lieu de destination de N0, N1 lieu source de N0, N1 lieu statique de N0, N1 lieu de passage de N0 lorsque le verbe est un verbe de mouvement (codage + pour la colonne V mouvement), ou par la colonne V statique dans le cas contraire.

6.3 Les classes des noms prédicatifs

Les classes des noms prédicatifs (cf. Tab. 5.1(b)) sont au nombre de 78 (ou 83 si l'on tient compte des tables de L. Pivaut). Les verbes supports sont principalement *avoir*, dont 29 classes sont concernées, et *faire*, dont 37 classes sont concernées (ou 42 si l'on tient compte des tables de L. Pivaut), même si d'autres verbes supports peuvent être codés dans les tables.

Nous allons détailler les modifications apportées aux constructions de base, comme nous l'avons fait dans Tolone *et al.* (2010). La plupart des modifications concernent la définition de la construction de base elle-même. Nous avons noté dans toutes les constructions, l'entrée nominale Det N, dont le nom est spécifié dans la colonne <ENT>N.

En ce qui concerne les noms, il est important de remarquer que les exemples sont absents des tables. Nous avons ajouté certains exemples qui figurait sur des fiches cartonnées[8]. Mais ces fiches étant loin d'être complètes, la plupart des noms figurent dans les tables sans savoir dans quel sens ils sont utilisés, notamment lorsqu'il y a des doublons, où seul leur codage peut les différencier.

Nous commençons par certains noms ayant le verbe support *avoir* (6.3.1), puis le verbe support *faire* (6.3.2). Ensuite, nous abordons l'étude de quelques constructions converses avec différents verbes supports, où certaines entrées adjectivales et verbales figurent également (6.3.3). Enfin, nous présentons les nouvelles tables ayant été numérisées récemment (6.3.4).

6.3.1 Verbe support avoir

Nous présentons 4 classes de J. Giry-Schneider et A. Balibar-Mrabti, 2 classes de J. Giry-Schneider, ainsi que 7 classes de A. Meunier, toutes les classes se construisant avec le verbe support *avoir*.

8. Nous avons réalisé ce travail en collaboration avec Wankawee Puangkor en 2009.

Tables AN07 à AN10

Certaines classes avaient des colonnes codant la construction de base (par exemple, N0 avoir Det N) avec différents déterminants possibles (par exemple, N0 avoir un N et N0 avoir le N). La construction de base était donc redéfinie à chaque fois, tout en spécifiant la nature du déterminant. Nous avons enlevé cette information redondante, en définissant une seule fois la construction de base, et en indiquant dans les colonnes concernées uniquement la nature du déterminant.

Ainsi, pour la table AN09, nous avons renommé les trois colonnes N0 avoir un N, N0 avoir un certain N et N0 avoir des N respectivement en Det =: un, Det =: un-certain et Det =: des, la construction de base définie dans Giry-Schneider et Balibar-Mrabti (1993) (p. 27) étant N0 avoir Det N (*Cette langue a un alphabet*).

Le même procédé a été employé pour la table AN10 décrite dans Giry-Schneider et Balibar-Mrabti (1993) (p. 30) avec comme construction de base N0 avoir Det N (*Max a des absences*), mais également dans d'autre tables comme AN07, AN06, ANDN, F21 et ANA, qui seront vues par la suite.

En ce qui concerne la table AN07, sa propriété définitoire est décrite dans Giry-Schneider et Balibar-Mrabti (1993) (p. 10) par N0 avoir Det N (*E+Modif), mais également la paraphrase N0 être de Det N (*E+Modif). Elle reconnaît :
*Ce monument a une architecture (*E+simple)*
*= Ce monument est d'une architecture (*E+simple)*
Mais, pour les mêmes raisons que précédemment, il est préférable de séparer les informations concernant la nature du déterminant de celles définissant la construction. C'est pourquoi nous avons défini la construction de base par N0 avoir Det N, avec Det =: un-Modif. Cela nous a permis d'ajouter également la propriété définitoire Det =: un-certain, puisque le déterminant *un certain* est compatible avec toutes les entrées (*Ce monument a une (*E+certaine) architecture*). La colonne N0 être de Det N Modif étant codée dans la table, il n'y a pas de deuxième construction définitoire. En effet, dans certains cas, cette construction ne s'applique pas :
Ce pays a une vieille culture
*= *Ce pays est d'une vieille culture*

La propriété définitoire de la table AN08 est décrite dans Giry-Schneider et Balibar-Mrabti (1993) (p. 17) par N0 avoir Det N = il y avoir Det N Loc N0, en précisant que *avoir* admet les variantes *comporter* et *comprendre*. Elle reconnaît :
Cette question (a+comporte+comprend) plusieurs aspects
= Il y a plusieurs aspects dans cette question
La colonne il y avoir Det N Loc N0 étant codée dans la table, nous ne l'avons pas gardée en tant que construction de base, puisqu'elle n'est pas acceptée par toutes les entrées :
Cette langue (a+comporte+comprend) une écriture
*= *Il y a une écriture (dans+de) cette langue*
Nous avons donc choisi d'avoir pour la table AN08 la construction de base N0 avoir Det N, avec les propriétés définitoires Vsup =: comporter et Vsup =: comprendre.

Tables ANSN et ANDN

La construction de base de la table ANSN est défini dans Giry-Schneider (2005b) (p. 220) par N0 avoir Det N (E+Qu P) sur N1 (*Max a un préjugé sur les femmes*). De plus, la colonne N0 avoir le N Qu P sur N1 est codée dans la table car elle n'est pas acceptée par toutes les entrées :
Max a sur les femmes le préjugé qu'elles conduisent mal
**Max a sur les femmes la lacune qu'elles conduisent mal*
C'est pourquoi la complétive ne doit pas figurer dans la construction de base, que nous avons notée N0 avoir Det N sur N1.

La construction de base de ANDN est défini dans Giry-Schneider (2005b) (p. 223) par N0 avoir Det N (de N1+Qu P) (*Max a une approche scientifique de cette question*). La complétive n'est possible que pour la moitié environ des noms (*Jean a conscience de la présence de Marie = Jean a conscience que Marie est présente*), nous avons donc également supprimé la complétive de la construction de base, notée N0 avoir Det N de N1.

Tables AN01 à AN06 et ANSY

Les tables AN01 à AN06 et ANSY de A. Meunier sont des nominalisations d'adjectifs avec le verbe support *avoir*. Ces 7 tables ont donc une double entrée : des adjectifs prédicatifs et leurs noms prédicatifs associés. Nous avons intégré ces tables dans les tables de noms prédicatifs, puisque c'est la seule catégorie réellement représentée jusqu'à présent. Ce qui nous a amené à intituler la colonne comportant l'entrée nominale <ENT>N, et la colonne comportant l'adjectif <OPT>Adj. Les propriétés définitoires concernent donc ces deux catégories, l'entrée nominale étant représentée par N et l'entrée adjectivale par Adj dans chacune des constructions concernées.

Trois constructions définitoires sont définies dans Meunier (1981) (p. 175) pour l'ensemble des tables (sauf la table AN06, cf. ci-dessous) :
N0 être Adj (E+Modif+Prép N1)
N0 avoir Det N (E+Modif+Prép N1)
N0 être de Det N (E+Modif+Prép N1)
Elles reconnaissent :
Jean est désinvolte
= *Jean a (une certaine+de la) désinvolture*
= *Jean est d'une grande désinvolture*
La construction de base adjectivale est la première et la construction de base nominale est la deuxième, avec *avoir* comme verbe support. C'est cette deuxième construction que nous retiendrons comme construction de base puisque nous considérons ici l'entrée nominale.

La table AN06 décrite dans Meunier (1981) (p. 205), se voit attribuer uniquement deux parmi les trois constructions définitoires précédentes :
N0 être Adj (E+Modif+Prép N1)
N0 avoir Det N (E+Modif+Prép N1)
*N0 être de Det N (E+Modif+Prép N1)

Elle reconnaît :
Cette table est poussiéreuse
= *Cette table a de la poussière*
= **Cette table est d'une certaine poussière*

Le problème est que cette notation comporte deux signes + par construction, ce qui signifie par exemple, que la construction N0 avoir Det N (E+Modif+Prép N1) équivaut aux trois constructions N0 avoir Det N, N0 avoir Det N Modif et N0 avoir Det N Prép N1. Or, elles ne sont pas toutes les trois acceptées par toutes les tables. Nous aboutissons donc à neuf constructions à traiter au cas par cas. N'ayant pas d'informations supplémentaires sur les propriétés définitoires, nous avons dû faire des approximations, aussi exactes que possibles, pour rendre compte de toutes les colonnes, en gardant une certaine cohérence entre elles. Les choix peuvent être discutables, puisque la meilleure solution serait un remaniement complet des tables, les déterminants, modifieurs, etc. n'étant pas forcément les mêmes pour les trois constructions principales. Il importe de préciser que nous ne prenons pas en compte les trois constructions avec Modif, car lors de la spécification du déterminant *un* la présence obligatoire d'un modifieur est indiquée, par la propriété Det =: un-Modif. Cette colonne est codée dans toutes les tables, sauf AN03 et AN04 où nous l'avons définie comme définitoire.

Par défaut, nous avons donc les six constructions suivantes définitoires pour toutes les tables :
N0 avoir Det N (construction de base possible)
N0 être de Det N
N0 être Adj
N0 avoir Det N Prép N1 (construction de base possible)
N0 être de Det N Prép N1
N0 être Adj Prép N1

À présent, nous montrons l'étude faite pour chaque construction une par une pour définir pour chaque table, celles que nous avons réellement gardées comme définitoires[9].

Une table nécessite un traitement spécifique, il s'agit de la table ANSY décrite dans Meunier (1981) (p. 202). Elle regroupe les paires à compléments symétriques, dont la définition implique qu'elles entrent dans les deux constructions suivantes :
N0 (être Adj+avoir Det N) Prép N1
= N0 et N1 (être Adj+avoir Det N)
Elle reconnaît :
Cet objet (est identique à+a une certaine identité avec) cet autre
= *Cet objet et cet autre (sont identiques+ont une certaine identité)*
Cela l'exclut de toutes les constructions sans Prép N1, et lui ajoute deux constructions définitoires supplémentaires :
N0 et N1 avoir Det N
N0 et N1 être Adj

9. Nous avons réalisé ce travail en collaboration avec Annie Meunier en 2010.

La construction de base est donc N0 avoir Det N Prép N1 et les autres constructions définitoires sont N0 être de Det N Prép N1, N0 être Adj Prép N1, N0 et N1 avoir Det N et N0 et N1 être Adj.

Examinons le cas des trois constructions sans complément, la table ANSY étant exclue :

– la construction N0 avoir Det N est la construction de base pour toutes les tables à l'exception de la table AN06, où la colonne est codée dans cette table, ce qui pose problème car elle n'a aucune construction de base nominale et lorsque la colonne est codée −, cela signifie que seule l'entrée adjectivale compte. Cette propriété est également codée dans les tables AN01 et AN02, mais les entrées codées − (par exemple, *Léa a une grande fécondité*) acceptent cette colonne au moins avec un modifieur, ce qui n'engendre aucun de problème puisque la colonne Det =: un-Modif est toujours codée + dans ces cas-là. C'est pourquoi, nous avons gardé N0 avoir Det N comme construction de base également pour les tables AN01 et AN02, elle l'est donc pour les tables AN01 à AN05 ;
– la construction N0 être de Det N n'existe pas car il y a obligatoirement un modifieur, elle est donc notée N0 être de Det N Modif (**Luc est d'une bonté*, à contraster avec *Luc est d'une bonté rare*). Cette construction est définitoire pour toutes les tables sauf pour AN06, qui est en fait une table de paires {N, N-a}, c'est-à-dire adjectivations de substantifs concrets. Cette table de paires {N, N-a} exclut la nominalisation N0 être de Det N Modif, qui n'est compatible qu'avec un N de caractère non comptable. Notons que c'est la seule construction que nous avons gardé avec la notation Modif, laquelle donne un sens à la colonne N0 être de Det N, qui est codée dans la table AN03, seule table acceptant parfois la construction sans le modifieur. La construction N0 être de Det N Modif est donc définitoire pour les tables AN01 à AN05 ;
– la construction N0 être Adj est le sujet même de la thèse de Meunier (1981), qui s'intitule *Nominalisation d'adjectifs par verbes supports*, elle ne figure donc dans aucune des tables et est définitoire des tables AN01 à AN06.

Voyons à présent les trois constructions avec le complément N1. Nous pouvons commencer par remarquer que la table AN04 contient les colonnes N0 avoir Det N pour N1 et N0 être Det N pour N1 (sans la préposition *de*) qui codent directement les constructions avec le complément N1 avec la préposition *pour*. De même, la table AN03, contenait la colonne Prép1 =: (avec+envers) que nous avons renommée et dupliquée en N0 avoir Det N (avec+envers) N1, N0 être de Det N (avec+envers) N1 et N0 être Adj (avec+envers) N1, afin de spécifier chaque construction complète avec les deux prépositions spécifiques, comme c'est le cas pour la table AN04. Cela justifie qu'il ne soit pas nécessaire d'établir de construction finissant par Prép N1 comme définitoire pour les autres tables, AN05 et AN06 ne faisant allusion à aucun complément dans leurs propriétés.

Les trois constructions avec le complément N1 ne concernent donc que les tables AN01, AN02 et ANSY :

– la construction N0 avoir Det N Prép N1 est la construction de base pour la table ANSY, comme nous l'avons évoqué précédemment. Elle contient la colonne

Prép1 (avoir Det N) codant la préposition associée à la construction, qui est codée pour toutes les entrées (sauf pour une entrée, mais cela a été corrigé). Ensuite, cette construction concerne les tables AN01 et AN02, ce qui se justifie par la présence de la colonne Prép1 codée dans la table. Le problème réside dans le fait que la préposition peut être vide et notée <*E*>, or cette notation est incorrecte car elle ne signifie pas que l'on peut accepter le complément N1 sans préposition, mais qu'il n'y a pas de complément N1. Nous avons donc remplacé tous les <*E*> par des −. Si l'on garde la construction définitoire N0 avoir Det N Prép N1, cela signifie que toutes les entrées acceptent un N1. C'est pourquoi nous l'avons ajoutée dans les tables AN01 et AN02 en la codant avec un + lorsqu'il y avait une préposition qui était codée et un − lorsqu'il n'y en avait pas. Cette construction n'est donc pas définitoire pour les tables AN01 et AN02 ;
- la construction N0 être de Det N Prép N1 est également définitoire de la table ANSY. Pour les tables AN01 et AN02, parallèlement à N0 avoir Det N Prép N1, la colonne a été ajoutée dans la table (avec le même codage que N0 avoir Det N Prép N1) et n'est donc pas définitoire des tables AN01 et AN02 ;
- la construction N0 être Adj Prép N1 ne figure pas explicitement, même si la préposition qui introduit le complément de la paire {Adj, Adj-n} (Adj-n désignant le nom issu de l'adjectif Adj) est codée. Mais par exemple, la conservation du complément pose souvent quelques problèmes. Nous savons que les paires à compléments sont regroupées en AN01 et AN02 et évidemment ANSY qui, par définition, a un complément symétrique du sujet. À défaut d'avoir plus d'information sur les tables AN01 et AN02, nous avons fait de même que pour N0 avoir Det N Prép N1 et N0 être de Det N Prép N1, nous avons ajouté la colonne N0 être Adj Prép N1 dans les tables, avec à nouveau le même codage (nous avons donc trois colonnes identiques), tout en sachant que les prépositions ne sont pas forcément exactes. En revanche, elle est définitoire de la table ANSY, sachant que les prépositions sont codées dans la table, puisqu'elle contient les deux colonnes Prép1 (être Adj) et Prép1 (avoir Det N). Cela permet de distinguer clairement les prépositions pour chacune des deux constructions.

Les trois constructions précédentes sont donc définitoires uniquement de la table ANSY.

Nous devons également tenir compte des entrées d'adjectifs prédicatifs simples en tant qu'entrées supplémentaires sans oublier les constructions définitoires définies avec Adj, même si le codage est beaucoup plus complet pour les noms prédicatifs. Nous ferons le bilan de ces entrées dans la section 6.6.

6.3.2 Verbe support faire

Nous présentons 31 classes de J. Giry-Schneider, puis, 15 autres classes de J. Giry-Schneider, qui toutes se construisent avec le verbe support *faire*[10].

10. Notons que la table F2B1 (contenant 65 entrées) est incluse dans la version 3.3 mais ne le sera pas dans la suivante puisqu'elle a été remplacée et annulée par la table FNDN. Cette dernière contient 136 entrées supplémentaires, mais 7 entrées ont également été supprimées (*générique*, *mise en scène*, *scénario*, *squelette*, *tissu*, *toile de fond*, *trame*) et ne figurent dans aucune autre table. De plus, même

Tables F1A à F91

Les tables F1A à F91 de J. Giry-Schneider, au nombre de 21, sont des nominalisations avec le verbe support *faire*, à partir des verbes déjà étudiés par M. Gross ou BGL, puisque leur classe est donnée dans les tables. Commençons par observer que cette colonne souffre d'un manque total de mise à jour, puisqu'elle ne tient pas compte de toutes les modifications récentes sur les verbes. En revanche, cela nous informe que ces verbes, dont la construction verbale est spécifiée, ne servent qu'à faire le lien avec les noms, mais ne sont en aucun cas à prendre en compte comme de nouvelles entrées verbales [11]. C'est pourquoi, nous ne détaillerons pas les modifications effectuées sur les constructions verbales, lesquelles font partie des constructions définitoires mais ne concernent pas les noms.

Toutes les constructions de base avec le verbe support *faire* sont définies dans Giry-Schneider (1978) (p. 28). Dans la majorité des cas, elles n'ont pas été modifiées, ou simplement séparées en plusieurs propriétés, ou encore simplifiées, car certaines colonnes sont codées dans la table :

- les tables F1A, F1B, F1C, F1D et F1R ont la construction de base N0 faire Det N (*Max fait de la navigation*) ;
- la table F6 a la construction de base N0 faire Det N de N1 Prép N2 (*Luc fait le change de ses dollars (contre+pour) des marks*) ;
- la table F8 a la construction de base N0 faire Det N Prép N1 Prép N2 (*Max fait une enquête sur cette affaire auprès de Marie*) ;
- les tables F4 et F41 ont la construction de base Nnr faire Det N à N1 (*Max a fait peur à Ève*). Dans un souci de réutilisabilité des intitulés, nous avons simplement renommé la construction de base N0 faire Det N à N1, avec N0 =: Nnr ;
- les tables F9 et F91 ont la construction de base Nnr faire le N de N1 (*Que Marie soit venue fait la surprise de Luc*). De la même manière que précédemment, nous avons renommé la construction de base N0 faire Det N de N1, avec N0 =: Nnr et Det =: le ;
- les tables F2A, F2B, F2B1, F2C et F2R ont la construction de base N0 faire Det N de N1 (E+à N2) (*Max fait l'extraction d'une dent (E+à Léa)*). Or, la colonne N0 faire Det N de N1 à N2 est codée dans la table car elle n'est pas acceptée par toutes les entrées (par exemple, *Max fait l'épuration des eaux usées*), c'est donc uniquement N0 faire Det N de N1 que nous avons gardé comme construction de base ;
- les tables F3 et F31 ont la construction de base N0 faire Det N (à+contre) N1. Les deux colonnes Prép1 =: à et Prép1 =: contre étant codées dans la table, nous avons

si la construction de base est la même, on passe de 20 colonnes codées dans la table à 7 colonnes uniquement : on perd notamment des constructions telles que N0 faire Det N de N1 à N2, ou avec la forme pronominale *se faire* (N0 se faire Det N de N1), ou encore des informations sur les réductions en groupe nominal (GN =: le N de N0). Le même cas de figure se présente avec la table F31 (contenant 102 entrées), qui a été remplacée et annulée par FNAN, avec 171 entrées supplémentaires et 27 entrées supprimées. Parmi celles-ci, 11 seulement figurent dans d'autres tables de J. Giry-Schneider (*baroud d'honneur* dans FNANN, *cuisine* dans FN), ou de G. Gross (*apothéose* dans FR2, *croc-en-jambe* dans FS3). C'est la raison pour laquelle la version 3.3 contient l'ensemble des tables à notre disposition.

11. Ces entrées ne sont pas considérées comme des entrées supplémentaires, comme c'est le cas dans d'autres tables (voir 6.3.3).

adopté pour la construction de base plus générale : N0 faire Det N Prép N1. Cela permet de reconnaître des entrées telles que :
Max fait de la diffamation contre Léa
Max fait du baratin à Ève

Trois tables ont nécessité plus d'attention, car nous avons dû généraliser certains intitulés figurant dans ces tables, sinon la construction de base ne pouvait être acceptée par toutes les entrées.

La table F21 a la construction de base N0 faire Det N (de N1+Qu P) (E+à N2), ce qui donnerait les quatre propriétés suivantes : N0 faire Det N de N1, N0 faire Det N de N1 à N2, N0 faire Det N Qu P et N0 faire Det N Qu P à N2. Or, les colonnes Prép N1 =: Qu Pind et Prép N1 =: Qu Psubj sont codées dans la table et reconnaissent :
Max a fait (E+à Ida) la déclaration qu'elle viendra avec une surprise
Max fait (E+à Marie) le reproche qu'elle ne veuille pas lui écrire
Habituellement, nous remplaçons une préposition par Prép dans une construction, lorsqu'une ou plusieurs colonnes codent explicitement les différentes prépositions possibles. Dans cette table, seule la préposition *de* est possible, mais cette préposition est effacée en présence de la complétive. De plus, la notation de N1 dans la construction de base, ne permet pas de faire le lien direct entre Prép N1 et Qu P. Nous avons donc comme constructions définitoires N0 faire Det N Prép N1 à N2, avec Prép1 =: de, et N0 faire Det N Prép N1. Elles reconnaissent :
Max a fait une escroquerie d'une somme importante
Max a fait l'escroquerie d'une somme importante (E+à Léa)
Mais toutes les entrées n'acceptent pas un N2, comme l'entrée *essai* (*Max fait l'essai de ce produit*). Cela est en fait codé par la colonne N0 faire Det N à N2, puisqu'elle sous-entend l'effacement possible de Prép N1, ainsi que la présence du complément N2. Étant le seul moyen de connaître les entrées acceptant un N2, nous avons dupliqué cette colonne pour coder de la même manière N0 faire Det N Prép N1 à N2. La construction de base est donc uniquement N0 faire Det N Prép N1.

La table F5 a la construction de base N0 faire Det N Prép N1 (E+de N2) où Prép =: à+dans+sur. Les prépositions sont bien codées dans la table, il est inutile de les préciser. Un exemple reconnu par cette table est :
Paul fait des broderies sur cette nappe (E+de jolies petites fleurs)
Ce cas est assez rare, et la majorité des entrées n'acceptent pas de N2 (*Max fait des éclaboussures sur le sol*). C'est la colonne N0 faire un N Prép N1 de N2 qui code la présence du N2, même si le déterminant n'est pas obligatoirement *un*, comme cela est indiqué dans la construction (cf. exemple de l'entrée *broderie*). De plus, les déterminants sont codés indépendamment des constructions (Det =: un, Det =: des, etc.). Nous avons donc renommé cette colonne N0 faire Det N Prép N1 de N2 et l'unique construction de base est N0 faire Det N Prép N1.

La table F7 a la construction de base N0 faire Det N entre N1 et N2 (*Max fait une distinction entre le vrai et le faux*). Or, cette colonne est codée dans la table, mais elle n'est pas toujours +. En effet, les constructions peuvent avoir des formes variées selon les verbes considérés (Giry-Schneider, 1978) (p. 246), il y a ainsi parfois plusieurs formes

pour un même verbe :
Jean fait le mélange du beurre avec la farine
Jean fait le mélange du beurre et de la farine
On peut ainsi dénombrer cinq constructions distinctes étant codées dans la table :
N0 faire Det N de N1 avec N2
N0 faire Det N de N1 et de N2
N0 faire Det N entre N1 et N2
N0 faire Det N de N1 et N2 entre eux
N0 faire Det N de N1 à N2
Il y a trop de disparités pour diviser la table et les noms n'acceptant pas la construction N0 faire Det N entre N1 et N2 sont discutables (par exemple, *addition*, *alliage*, *substitution*, etc.). C'est pourquoi, N0 faire Det N entre N1 et N2 reste la construction de base. Il faut souligner le fait que cette généralisation a été faite afin de pouvoir faire la conversion au format Le*fff* (voir section 8.1) qui nécessite une construction de base pour chaque table.

Tables FN à FNPNN

Les tables FN à FNPNN de J. Giry-Schneider sont des constructions nominales avec le verbe support *faire*, elles représentent 10 classes, et sont de trois sortes :
- celles se terminant par N : elles sont composées uniquement d'un nom (<ENT>N) ;
- celles se terminant par NA : elles sont composées d'un nom et d'un adjectif (<ENT>N et <ENT>Adj), dont l'ordre est interverti si la colonne Adj permut obl est codée + dans la table ;
- celles se terminant par NN : elles sont composées d'un nom, d'une préposition, éventuellement d'un déterminant (souvent vide) et d'un deuxième nom (<ENT>N, <ENT>Prép, <ENT>Detc et <ENT>Nc).

Elles sont ensuite divisées en quatre ensembles.

Pour commencer, examinons l'ensemble des tables dont le nom commence par FN dont la construction de base décrite dans Giry-Schneider (1987) (p. 97) est N0 faire Det N :
- table FN : *Max fait des abdominaux* ;
- table FNA : *Max a fait une faute professionnelle* (*Max a fait une bonne affaire*, avec l'adjectif interverti) ;
- table FNN : *Max a fait une erreur de jeunesse* (*Max fait la course contre la montre*, avec un déterminant non vide).

Puis, observons l'ensemble des tables dont le nom commence par FNPN dont la construction de base décrite dans Giry-Schneider (1987) (p. 107) est N0 faire Det N (E+Prép N1), que nous avons simplement séparée en deux constructions : N0 faire Det N Prép N1 comme construction de base et N0 faire Det N comme deuxième construction définitoire :
- table FNPN : *Luc a fait un colloque (E+sur ce thème)* ;
- table FNPNA : *Luc a fait une coupe sombre (E+dans son manuscrit)* ;
- table FNPNN : *Max a fait un article de fond (E+sur ce sujet)*.

Certains de ces noms composés sont déjà présents dans une autre table en tant que nom simple, par exemple *article de fond* (FNPNN) et *article* (FNPN). C'est pourquoi nous ne

pouvons pas considérer uniquement le premier nom de ces noms composés comme entrée de la table également, d'autant plus que, souvent, plusieurs noms composés commencent par le même nom simple, on aurait alors beaucoup de doublons de la même entrée. En revanche, tous les mots simples contenus dans les mots composées ne sont pas tous codés, par exemple *erreur de jugement* (FNPNN) alors que *erreur* n'est codé dans aucune table, il faudrait donc les ajouter. En effet, la phrase suivante est tout a fait acceptable :
Max a fait une erreur (E+de jugement) sur cette affaire

Ensuite, considérons l'ensemble des tables dont le nom commence par FNAN dont la construction de base décrite dans Giry-Schneider (1987) (p. 125) est N0 faire Det N à N1, avec N0 =: Nhum et N1 =: Nhum obligatoires (et donc N0 =: N-hum et N1 =: N-hum codées −). Le complément N1 étant tout aussi facultatif que pour les tables FNPN, nous avons également ajouté la deuxième construction définitoire N0 faire Det N :

- table FNAN : *Max fait un canular (E+à Marie)* (cf. Fig. 3.4) ;
- table FNANA : *Max fait les yeux doux (E+à Marie)* ;
- table FNANN : *Léa a fait le coup de la panne (E+à Jean).*

Enfin, la table FNDN a comme construction de base N0 faire Det N de N1 décrite dans Giry-Schneider (1987) (p. 133) (*Max a fait la biographie de Luc*). Pour cette table, le complément N1 est obligatoire. De plus, la colonne N0 faire un N, codée dans la table, permet de reconnaître certaines entrées avec un sens différent :
Max fait le film des événements (film = «récit»)
Max fait un film (film = «cinéma»)

6.3.3 Constructions converses

Les tables AA à IS2 (ou plus exactement : AA, AD, DR1 à DRC, ES, FR1 à FR3, FS1 à FS3, IS1 et IS2) de G. Gross décrites dans Gross (1989) constituent 15 tables de noms prédicatifs avec des constructions converses entre deux verbes supports qui varient en fonction des tables.

Les tables DR1, DR2 et DR3 (Donner-Recevoir) ont N0 donner Det N à N1 comme construction de base et N1 recevoir Det N de (E+la part de) N0 comme construction converse (p. 57, 129, 189, 216). Nous avons renommé cette deuxième construction définitoire en N1 recevoir Det N (de+de la part de) N0, pour identifier correctement les deux prépositions qui peuvent être *de* ou *de la part de*. Cette notation a été changée pour l'ensemble des tables. Illustrons ces deux constructions :
Max a donné (une+des+quelques) claque(s) à Luc
= Luc a reçu (une+des+quelques) claque(s) (de+de la part de) Max

La table DR1 contient des nominalisations à partir de verbes, dont l'entrée nominale liée morphologiquement est notée <ENT>N et l'entrée verbale <OPT>V :
Luc a ordonné à Max de se taire
= Luc a donné l'ordre à Max de se taire
Remarquons que le deuxième complément est identique dans la construction verbale et dans la construction à verbe support, mais qu'aucune des deux n'est codée dans la table. Notons également que ces verbes ne sont pas obligatoirement présents dans les tables de verbes, car ils sont d'un usage assez ou relativement rare, comme par exemple le verbe

semoncer associé au nom *semonce* (table DR1), qui signifie *réprimander*. Le problème réside dans le fait qu'il n'est pas possible d'établir une construction de base verbale, car elle peut être intransitive ou transitive en fonction des entrées, comme par exemple (en opposition à l'entrée *ordonner*) :
Luc a appuyé ce projet
= Luc a donné son appui à ce projet
Il faudrait donc coder les deux colonnes N0 V à N1 et N0 V N1 [12] dans la table pour les verbes absents des tables de verbes (ou encore mieux les ajouter en tant qu'entrées dans les tables de verbes pour les coder). Puis, pour les verbes présents, il faudrait faire référence à l'identifiant du verbe dans les tables de verbes et coder – (ou supprimer) les deux propriétés précédentes (ce qui permettrait de ne pas en tenir compte comme entrée verbale). Actuellement, afin de pouvoir tenir compte de ces nouvelles entrées verbales, les deux constructions sont temporairement considérées comme définitoires (on reconnaît donc des phrases agrammaticales). Cela a également été réalisé pour les tables FR1, FS1 et IS1.

La table DR2 comprend des noms prédicatifs non reliés à un verbe (p. 132), comme par exemple *attention* et la table DR3 contient les noms composés (p. 139), comme par exemple *délégation de pouvoir* dont l'entrée est entièrement contenue dans <ENT>N, contrairement aux tables FN à FNPNN de J. Giry-Schneider. D'autres verbes supports sont acceptés et sont codés dans la table, avec des propriétés telles que donner = accorder ou recevoir = avoir, faisant référence à la construction standard ou converse. La numérotation des tables a la même signification dans les tables suivantes : 1 pour déverbaux, 2 pour autonomes et 3 pour composés.

La table DRC (Donner-Recevoir un Coup) a N0 donner un coup de N à N1 comme construction de base et N1 recevoir un coup de N (de+de la part de) N0 comme construction converse (p. 151, 216) :
Max a donné un coup de bâton à Luc
= Luc a reçu un coup de bâton (de+de la part de) Max
Cette table contient aussi bien des noms déverbaux (V-n) que des noms isolés (N), c'est-à-dire non reliés à un verbe. Ils se distinguent dans la table par la colonne <OPT>V contenant un verbe ou un – (comme par exemple, le nom *bâton* relié à *bâtonner*, et le nom *bambou* non relié à un verbe). Ici, la construction verbale est toujours N0 V N1, mais elles est codée dans la table uniquement lorsque le verbe est présent.

Les tables FR1, FR2 et FR3 (Faire-Recevoir) ont N0 faire Det N Prép N1 comme construction de base et N1 recevoir Det N de (E+la part de) N0 (p. 218) comme construction converse :
Max fait des flatteries à Léa (table FR1)
Léa reçoit des flatteries de la part de Max
Les entrées nominales de ces tables sont extraites des tables F1A à F91 de Giry-Schneider (1978) pour les noms reliés à un verbe (plus exactement, des tables F2, F3, F5 et F6 pour la table FR1) et des tables FN à FNPNN de Giry-Schneider (1987) pour les noms isolés (plus exactement, de la table FNAN pour la table FR2 et des tables FNDN, FNANA

12. Cela revient à accepter la disjonction de propriétés suivante : (N0 V à N1) ou (N0 V N1).

et FNANN pour la table FR3). Quelques noms ont été ajoutés dans chacune des tables. Par exemple, dans la table FR1, il y a des noms tels que *commande*, *exhortations*, *interdiction*, *menaces* et *obligation*. Dans la table FR2, contrairement à FNAN, des noms dont le complément peut être un nom non humain ont été ajoutés. Dans la table FR3, il y a également des noms composés nouveaux. La difficulté réside donc dans le fait de pouvoir les reconnaître pour pouvoir associer les entrées qui sont en double avec l'entrée originale des tables de J. Giry-Schneider.

Les tables FS1, FS2 et FS3 (Faire-Subir) ont N0 faire Det N Prép N1 comme construction de base et N1 subir Det N de (E+la part de) N0 comme construction converse (p. 262) :
Max a fait une contre-expertise de ce document (table FS3)
Ce document a subi une contre-expertise de la part de Max

Les tables IS1 et IS2 (Infliger-Subir) ont N0 infliger Det N à N1 comme construction de base et N1 subir Det N de (E+la part de) N0 comme construction converse (p. 282) :
Max a infligé une défaite à Luc (table IS2)
Luc a subi une défaite de la part de Max
L'effectif étant réduit, quelques noms composés ont été ajoutés à IS2.

La table ES (Exercer-Subir) a N0 exercer Det N Prép N1 comme construction de base et N1 subir Det N (de+de la part de) N0 comme construction converse (p. 285) :
Max exerce son arrogance contre Luc
Luc subit l'arrogance de Max
Les noms pouvant être des V-n, des N ou des Adj-n, les deux colonnes <OPT>V et <OPT>Adj figurent dans la table et peuvent être vides toutes les deux. Par exemple, le nom *attirance* est associé au verbe *attirer* et à l'adjectif *attirant*. Il faudrait vérifier que les verbes figurent bien tous dans les tables de verbes, et les adjectifs peuvent être utilisés pour créer des entrées adjectivales puisqu'il n'y a pas de tables d'adjectifs utilisables actuellement. De plus, les colonnes N0 V N1 et N0 être Adj Prép N1 sont codées lorsque le verbe et/ou l'adjectif sont présents.

La table AA (Avoir-Avoir) a N0 avoir Det N Prép N1 comme construction de base et N1 avoir Det N de N0 comme construction converse (p. 233) [13]. Les noms de cette table sont soit des nominalisations d'adjectifs extraits des tables AN01 à AN06 et ANSY de Meunier (1981), soit des noms isolés, ou soit des noms déverbaux correspondant pour la plupart du temps aux verbes de la table 12 (ou de la table 6) de Gross (1975). Les verbes et adjectifs n'étant pas codés dans la table, ils ne seront donc pas pris en compte. Il faudrait donc repérer quels sont les noms déjà codés dans les tables de A. Meunier pour pouvoir les fusionner en définissant une priorité d'une des deux constructions sur l'autre et en ajoutant des colonnes à la table correspondante.

Prenons un exemple, l'entrée *bienveillance* figure dans la table AA de G. Gross, mais est aussi dans la table AN03 de A. Meunier, et même également dans la table APE21 de Vivès (1983). Le sens est le même pour ces trois tables :
Max a une certaine bienveillance (pour+envers+à l'égard de) Luc
On peut observer des différences dans le codage des verbes supports, des déterminants

13. Pour la table AA, la construction converse au sens de Gross (1989) est analysée comme construction à *opérateur à lien* par Danlos (1988) (p. 27).

et des constructions inverses ou converses. Examinons l'exemple suivant, aussi complet que possible, illustrant les différentes propriétés de cette entrée, dans la table AA :
Max (a+ressent+éprouve) (une grande+de la) bienveillance (pour+à l'égard de) Luc
Luc (a+est l'objet de+fait l'objet de) une grande bienveillance de Max
Dans la table AN03 :
Max a (une grande+de la) bienveillance (avec+envers) Luc
Dans la table APE21 :
Max a (la bienveillance de Luc+sa bienveillance)
Max perd ((la+une) bienveillance de Luc+sa bienveillance)
Luc a (une grande+de la) bienveillance pour Max
Luc perd toute bienveillance pour Max
Luc (est de+est sans) Det bienveillance pour Max[14]
Aucune entrée n'est complètement identique à l'autre, et les entrées nominales ne sont pas toutes pourvues d'exemple, ce qui permettrait d'identifier leur sens. Nous rencontrons donc un réel problème pour reconnaître que ces trois entrées sont à fusionner, sans pour autant savoir ensuite comment les fusionner.

La table AD (Avoir-Donner) a comme construction de base uniquement N0 donner Det N à N1 (p. 27), qui est la construction la plus longue (avec un N1, permettant de faire le lien lorsqu'il passe en position sujet). La colonne N1 avoir Det N est codée dans la table :
Le visage de Marie a des rides
Le temps a donné des rides au visage de Marie
Le temps a ridé le visage de Marie
Les noms sont tous déverbaux et la colonne <OPT>V est codée, ils sont donc à garder. De plus, la colonne N0 V N1 est codée + pour toutes les entrées verbales.

On peut donc noter comme entrées supplémentaires, les adjectifs prédicatifs simples de la table ES, et les verbes simples des tables DR1, DRC, FR1, FS1, IS1, ES et AD n'étant pas déjà codés dans les tables verbales. Ces entrées, dont le nombre est donné dans la section 6.6, restent évidemment à vérifier pour éviter de créer trop de doublons. De plus, notons que dans les tables DRC, ES et AD, les constructions verbales et adjectivales sont codées. Elles ne sont pas définitoires, comme c'est le cas pour les tables AN01-AN06 et ANSY. Cela permet de les coder − lorsqu'aucun verbe ou adjectif n'est associé à l'entrée. Cependant, la construction verbale des tables DR1, FR1, FS1 et IS1 reste à coder selon les entrées, afin d'éviter d'accepter des constructions agrammaticales.

6.3.4 Nouvelles classes

Considérons à présent les quatre ensembles de tables que nous avons numérisées récemment : 2 classes de D. de Négroni-Peyre avec le verbe support *être en*, 6 classes de

14. Notons que les déterminants possibles ne sont pas codés pour les deux constructions N1 être de Det N Prép N0 et N1 être sans Det N Prép N0, et que les déterminants acceptés par *avoir* et *perdre* dans les constructions converses ne semblent pas tous acceptables (*une grande+de la+toute*).

R. Vivès et 8 classes de J. Labelle avec le verbe support *avoir*, et 5 classes de L. Pivaut avec le verbe support *faire*.

Tables PSY et SYM

Les tables PSY et SYM de D. de Négroni-Peyre sont des nominalisations avec le verbe support *être en*, à partir des verbes déjà étudiés par M. Gross ou BGL puisque leur classe est donné dans les tables. C'est une étude des paires {V, V-n}, V-n que nous avons renommées N dans les constructions. D'ailleurs, le problème est que seules les entrées verbales sont codées dans la colonne <ENT>V (et non <OPT>V car pour le moment, il s'agit de l'entrée principale), alors que pour trouver le nom, seul le suffixe à ajouter au verbe est codé dans la colonne suffixe -n. Nous ne devons pas prendre en compte ces verbes comme de nouvelles entrées verbales, il faudrait donc écrire explicitement chaque nom afin de pouvoir les prendre en compte comme des noms prédicatifs [15]. De plus, ces tables sont décrites dans de Négroni-Peyre (1978) mais aucune construction de base n'est spécifiée. En effet, elles sont définies p. 136 comme étant les constructions *psychologiques* associées à la préposition *devant* pour la table PSY :
Marie rage devant ta méchanceté
= *Marie est en rage devant ta méchanceté*
Ainsi que les constructions *symétriques* qui mettent en jeu la préposition *avec* pour la table SYM :
Mic cohabite avec Flo
= *Mic et Flo cohabitent*
= *Mic est en cohabitation avec Flo*
= *Mic et Flo sont en cohabitation*
Les différentes prépositions possibles sont codées dans les tables. Les relations entre verbes et noms sont décrites de la sorte (p. 131), en sachant que le verbe peut être pronominalisé en se V :
N0 V N1 = N0 être en N Prép N1
N0 V Prép N1 = N0 être en N Prép N1
N0 V N1 Prép N2 = N1 être en N Prép N2

La construction N0 être en N Prép N1 étant la seule effectivement partagée par les deux tables (le complément N2 ne concerne que les entrées de la table SYM ayant la colonne N0 V N1 avec N2 codée +), et ne variant que par sa préposition, nous l'avons établie comme définitoire des deux tables. Puis, nous avons renommé les colonnes de la table PSY : N0 être en N contre N1, N0 être en V-n après N1 et N0 être en V-n devant N1 en Prép1 =: contre, Prép1 =: après et Prép1 =: devant ; ainsi que celles de la table SYM : N0 être en N avec N1, N0 être en N contre N1 et N0 être en N à N1 en Prép1 =: avec, Prép1 =: contre et Prép1 =: à. Pour la table SYM, nous avons également défini la construction définitoire N0 et N1 être en N pour rendre compte de la symétrie. Nous n'avons pas établi de constructions définitoires pour les verbes car les constructions verbales sont

15. Il faudrait saisir les noms comme cela a été fait par Eric Laporte pour les tables de nominalisation AN01 à AN06 et ANSY d'A. Meunier et F1A à F91 de J. Giry-Schneider, dans lesquels seul l'adjectif ou le verbe était en clair, comme on peut le voir dans les ouvrages originaux.

codées dans les tables.

Tables APE1 à APP3

Les tables APE1 à APP3 de R. Vivès décrites dans Vivès (1983) (p. 164) sont au nombre de 9. Par définition, les tables dont le nom commence par APE ont comme verbe support *avoir* et *perdre*, alors que les tables dont le nom commence par APP ont comme verbe support *avoir*, *perdre* et *prendre*. Les tables sont divisées en trois ensembles en fonction de la présence ou non de compléments :

– les tables numérotées 1 n'ont aucun complément. La construction de base de la table APE1 est N0 (avoir+perdre) Det N :
  Luc a de l'abnégation
  Luc a perdu toute abnégation
  Pour ne pas créer de nouveaux intitulés, nous avons réutilisé les intitulés déjà existants pour le verbe *avoir*, puis rajouté les autres verbes supports, et ceci pour toutes les constructions de base des tables. Pour la table APE1, la construction de base actuelle est donc N0 avoir Det N, avec Vsup =: perdre. De plus, il est spécifié p. 171 que pour l'ensemble de la table, *garder* et *conserver*, extensions aspectuelles de *avoir*, sont acceptées. C'est pourquoi nous avons ajouté les propriétés Vsup =: garder et Vsup =: conserver définitoires de la table.
  Pour la table APP1, la construction de base est N0 avoir Det N, avec Vsup =: prendre et Vsup =: perdre :
  Luc (a+prend) (de l'+un certain) allant
  Luc a perdu tout allant
  À cela s'ajoute la table APE11 qui accepte également N0 avoir Det N comme construction de base, avec Vsup =: perdre et N0 =: Nhum, le nom étant psycho-physiologique :
  Luc (a+a perdu) le (moral+forme) ;
– les tables numérotées 2 ont un seul complément, que nous avons noté N1 et non N2 comme à l'origine, puisque le nom n'est pas compté comme premier complément dans l'ensemble des tables nominales. La construction de base de la table APE2 est donc N0 avoir Det N Prép N1, avec Vsup =: perdre, la préposition étant codée dans la colonne Prép1 :
  Luc a des accès officiels auprès de Marie
  Luc a perdu tout accès auprès de Marie
  Celle admise par la table APP2 est N0 avoir Det N Prép N1, avec Vsup =: prendre et Vsup =: perdre :
  Luc (a+prend) une certaine emprise sur (Marie+la vente du cuir)
  Luc a perdu toute emprise sur (Marie+la vente du cuir)
  Doit également être prise en compte la table APE21 qui a pour construction de base N0 avoir Det N de N1, avec Vsup =: perdre et N1 =: Nhum, mais aussi la construction *inverse*, ou plutôt converse, N1 avoir Det N pour N0 [16], avec Vconv =: perdre, la notation Vconv désignant le verbe support converse, et permettant de différencier les

16. Nous n'avons pas gardé le terme de construction inverse employé dans Vivès (1983), malgré le

verbes supports de chaque construction [17]. Nous avons renommée cette construction définitoire N1 avoir Det N Prép N0 car la préposition Prép0 est codée lexicalement dans la table :
Luc (a+perd) l'estime de Max
Max a de l'estime pour Luc
Max perd toute estime pour Luc

N'oublions pas la table APE22 qui accepte cette même construction de base N0 avoir Det N de N1, avec Vsup =: perdre et N1 =: Nhum, mais également une construction *converse* qui n'est pas décrite et que nous avons définie par N1 Vconv Det N à N0, le verbe support converse Vconv étant codé dans la table par les quatre colonnes nommées Vconv =: accorder, Vconv =: adresser, Vconv =: donner et Vconv =: prêter :
Luc (a+perd) l'appui de Max
Max (donne+accorde) (un immense+son) appui à Luc
De plus, il est spécifié (p. 177) que les extensions aspectuelles *garder* et *conserver* sont régulières. Nous avons donc ajouté les propriétés Vconv =: garder et Vconv =: conserver définitoires de la table ;

– les tables numérotées 3 ont également un seul complément mais celui-ci est facultatif, et cela pour toutes les entrées. La table APE3 a donc comme construction de base N0 avoir Det N (E+Prép N1), avec Vsup =: perdre. Afin de supprimer le +, nous avons séparé cette construction en deux constructions : N0 avoir Det N Prép N1 comme construction de base et N0 avoir Det N Prép N1 comme deuxième construction définitoire :
Luc a un certain acharnement (E+au travail)
Luc a perdu tout acharnement (E+au travail)
De même, la table APP3, accepte N0 avoir Det N Prép N1 comme construction de base et N0 avoir Det N comme construction définitoire (au lieu de N0 avoir Det N (E+Prép N1), avec Vsup =: prendre et Vsup =: perdre :
Luc (a+prend) un rôle important (E+dans la société)
Luc a perdu son rôle important (E+dans la société)

---

fait que les deux constructions soient inversées par rapport à celles de la table AA. Notons de plus que la table APE21 contient pratiquement les mêmes entrées que la table AA, il serait donc judicieux de garder la table AA en y ajoutant les informations de la table APE21. En effet, Gross (1989) ayant etudié toutes les converses du français, sa terminologie est plus cohérente avec ses autres classes.

17. Notons que pour les constructions converses des tables AD à IS2 de G. Gross, la notation Vconv n'est pas utilisée car les colonnes s'intitulent, par exemple pour la table IS2, infliger = administrer pour les verbes supports de la construction de base ou subir = écoper pour les verbes supports converses. Cependant, elle apparaît dans le lexique *LGLex*, au même titre que Vsup puisque cela permet d'avoir deux listes distinctes de verbes supports, chacune se référant à l'une des deux constructions. En réalité, toutes ces colonnes devraient être renommées de la même manière que la table APE21, car cela peut engendrer un problème pour les constructions converses telles que AA, dont il est difficile de voir que la propriété avoir = ressentir concerne le verbe support de la construction de base, alors que la propriété avoir = subir concerne le verbe support converse. La distinction se fait actuellement car les colonnes sont placées à droite pour les Vsup et à gauche pour les Vconv dans les tables et aucune n'est ambiguë.

Tables ANA à ANSU

Les tables ANA à ANSU (sauf ANDN et ANSN) de J. Labelle décrites dans Labelle (1974) (p. 14) sont des noms prédicatifs avec le verbe support *avoir*, c'est la raison pour laquelle les 8 tables commencent par AN (Avoir N).

La table ANS (Symétrique) a pour construction de base N0 avoir Det N avec N1 (p. 36) et la construction symétrique N0 et N1 avoir Det N (*Max a eu une aventure avec Marie / Max et Marie ont eu une aventure*). En réalité, le verbe support peut être *avoir* et/ou *avoir eu*, ce qui est codé par les deux colonnes Vsup =: avoir et Vsup =: avoir eu (l'exemple précédent, à contraster avec *Max a une amitié avec Marie*). C'est pourquoi nous avons renommé les deux constructions en N0 Vsup Det N avec N1 pour la construction de base, et N0 et N1 Vsup Det N pour la deuxième construction définitoire, le Vsup permettant de faire le lien avec les deux colonnes. Nous avons également renommé tous les verbes supports avoir en Vsup dans les constructions codées dans la table. Cela a également été fait pour l'ensemble des tables. Il est important de souligner que dans les autres tables de noms prédicatifs, la propriété Vsup =: avoir implique également Vsup =: avoir eu, puisque le verbe support peut être conjugué à tous les modes et à tous les temps. C'est pourquoi, lors de la conversion, la distinction entre *avoir* et *avoir eu* n'est pas faite pour le moment, même si elle est faite dans les tables ANA à ANSU.

La table ANA (constructions avec Prép =: à) a pour construction de base N0 avoir Det N à ce Qu P (p. 85), que nous avons renommée N0 Vsup Det N Prép N1, avec Prép1 =: à, car la préposition peut avoir d'autres valeurs (*Ce liquide a une grande capacité (à+pour) se répandre / Ce vêtement a une bonne étanchéité (à+contre) la pluie*). De plus, la complétive n'est pas obligatoire puisqu'elle est codée par les deux colonnes N1 =: Qu Pind et N1 =: Qu Psubj, qui peuvent être toutes les deux codées −. Le Vsup peut être ici *avoir* et/ou *avoir eu*, mais aussi *manifester* et/ou *donner* (*Ce vêtement (a+manifeste+donne) une bonne étanchéité (à+contre) la pluie*), même si le verbe *avoir* est codé + pour toutes les entrées. Le verbe *avoir* est d'ailleurs codé + pour toutes les entrées de la majorité des tables suivantes (sinon cela est précisé).

La table ANML (substantifs de Maladie avec complément Locatif) a pour construction de base N0 avoir Det Nmal Loc Npc (p. 121) qui provient en fait de deux constructions : N0 avoir Det N Loc Npc prévue pour la table ANML (*Max a une éraflure au bras*) et N0 avoir (mal+une éraflure) Loc Npc qui devait définir la table ANPC, mais qui n'existe pas. C'est pourquoi nous avons retenu N0 Vsup Det N Loc N1pc comme construction de base, après avoir numéroté le complément N1 pour faire le lien avec la préposition locative Loc1 codée dans la table. Nous avons également renommé le Nmal en N, car le fait que les entrées désignent des noms de maladie n'a pas besoin de figurer dans toutes les constructions, qui ont donc été toutes renommées. Ceci a été réalisé également pour les trois tables suivantes. D'autre part, la propriété sémantique N =: Nmal a été ajoutée comme propriété définitoire, afin de ne pas perdre cette information. Le problème de la table ANML réside dans le fait que la construction de base est également une colonne qui est codée dans la table sans être toujours codée +. N'ayant pas trouvé le sens de cette colonne, elle est ignorée pour accepter la construction de base pour toutes les entrées. Dans cette table et dans la suivante, le verbe *avoir* est également codé + pour toutes

les entrées, même si pour cela nous avons dû forcer certaines entrées qui n'acceptaient aucun verbe support.

La table ANM (substantifs de Maladie sans compléments locatif) a pour construction de base N0 avoir Det N (p. 121) (Nmal étant sous-entendu comme le témoignent les colonnes de la table) que nous avons renommée N0 Vsup Det N, avec N =: Nmal (*Max a de la fièvre*).

La table ANMR (substantifs ne se construisant qu'avec *avoir eu*) a pour construction de base N0 avoir eu Det N (E+Loc Npc) (p. 121) (avec Nmal sous-entendu), que nous avons renommée N0 Vsup Det N, avec N =: Nmal (*Max a eu un soufflet (E+sur la joue)*). En effet, la colonne Loc N1pc code la possibilité d'ajouter un complément locatif à la construction de base. De plus, on avait la colonne Loc Npc =: <E> qui codait la facultativité de ce substantif, et qui avait le même codage que Loc N1pc (sauf pour une entrée sans que cela ne soit justifié). Cela signifie pour une entrée que, soit elle n'accepte pas de complément locatif, soit elle en accepte un mais qu'il est obligatoirement facultatif, ce qui justifie notre choix de mettre la construction de base sans locatif. La colonne Loc Npc =: <E> est donc ignorée puisque la facultativité est déjà codée dans la construction de base. Nous pouvons remarquer que la notation Vsup a été gardée pour être conforme aux autres tables, bien que Vsup =: avoir eu soit l'unique colonne figurant dans la table (elle est codée + pour toutes les entrées). De plus, la propriété Vsup =: avoir n'étant jamais acceptée, elle est codée −.

La table ANSU (constructions avec Prép =: sur) a pour construction de base N0 avoir Det N sur N1 (p. 156), que nous avons renommé N0 Vsup Det N sur N1 (*Max a un certain ascendant sur Léa*). Les constructions codées dans la table contenaient également des Nmal alors que les entrées ne sont pas des noms de maladie (la propriété N =: Nmal n'est donc pas définitoire).

La table ANP (constructions avec Prép =: pour) a pour construction de base N0 avoir Det N pour N1 (p. 169), que nous avons renommée N0 Vsup Det N pour N1 (*Marie a des dispositions pour la peinture*). Dans cette table et la suivante, le Vsup est, soit *avoir*, soit *avoir eu*, sans faire l'unanimité. Nous avons dû forcer trois entrées de cette table qui n'avait aucun verbe support, à accepter le verbe *avoir*.

La table ANPR (constructions avec autres Prép) a pour construction de base N0 avoir Det N Prép N1 avec Prép != avec+à+sur+pour (p. 177), que nous avons renommée N0 Vsup Det N Prép N1 (*Max a une dent contre Luc*). La préposition est codée par une des six colonnes Prép =: envers, Prép1 =: contre, Prép1 =: auprès de, Prép1 =: entre, Prép1 =: devant, Prép1 =: Loc. Nous considérons donc les quatre propriétés suivantes fausses pour l'ensemble des entrées : Prép1 =: avec, Prép1 =: à, Prép1 =: sur, Prép1 =: pour.

Tables FD1 à FD4

Les tables FD1 à FD4 de L. Pivaut[18] décrites dans (Pivaut, 1989) (p. 64) regroupent des noms prédicatifs avec le verbe support *faire*, qui se réfèrent à une activité musicale, sportive ou intellectuelle. Les tables sont au nombre de 5 et commencent par

18. Signalons que ces tables étant les dernières a avoir été numérisées, elles ne font pas partie de la version 3.3 mais seront incluses dans la suivante (Tolone, 2011).

FD (Faire Det N). La construction de base acceptée par les 5 tables est : N0 faire Det N (p. 1), avec N0 =: Nhum (p. 36) et Det =: du, faire = pratiquer (p. 45), qui signifie que lorsque le déterminant est partitif la valeur sémantique de la phrase est *pratiquer* (*Max (fait+pratique) du sport de combat*).

Au niveau syntaxique, les tables se distinguent par (p. 64) :

– l'impossibilité de l'article indéfini *un* en position Det pour la table FD4 :
  *Max fait (de l'aérobic+*un aérobic)*
  Nous avons codé − les deux propriétés Det =: un, faire = fabriquer et Det =: un, faire = pratiquer ;
– l'impossibilité d'un adverbe de temps duratif quand *un* est sélectionné pour les tables FD2, FD3A et FD3B :
  *Max a fait (du football pendant un an+*un football pendant un an)* (table FD2)
  Nous avons ajouté la propriété Det =: un, Advtd (Advtd représentant un adverbe de temps duratif) en la codant − pour les tables FD2, FD3A et FD3B, mais également FD4 alors qu'elle est codée + pour la table FD1. De plus, Dnum en position Det est très contraint, mais cette information n'est pas exploitable (codage O de la propriété Det =: Dnum) ;
– en plus du point précédent, le groupe nominal prédicatif contient toujours un déterminant numéral suivi d'une unité de distance pour les tables FD3A et FD3B : l'entrée est de la forme <ENT>Dnum <ENT>N (*Max fait du cinq mille mètres*) pour la table FD3A et <ENT>Dnum <ENT>N <ENT>Nc <ENT>Adjc (*Max fait du cent mètres nage libre*) pour la table FD3B (FD3A et FD3B ne représentent pas des différences de construction, mais des différences de morphologie des entrées) ;
– le fait que *faire* devient synonyme de *fabriquer*, *faire* apparaît comme une extension aspectuelle de *avoir*, *avoir un* est substituable à *faire un* et Dnum est partout possible en position Det pour la table FD1 :
  Max fait du piano / Max (fait+fabrique+a) (un piano+dix pianos)
  Nous avons codé + les propriétés Det =: un, faire = fabriquer et Det =: un-Modif, faire = fabriquer alors que les propriétés Det =: un, faire = pratiquer et Det =: un-Modif, faire = pratiquer sont codées −. Nous avons ajouté la propriété Det =: un, faire = avoir en la codant + pour la table FD1 et − pour les autres. Enfin, nous avons codé + la propriété Det =: Dnum.

Cela peut se résumer par le fait que si l'ensemble des propriétés suivantes est vérifié, il s'agit d'une entrée de la table FD1 :
Det =: un, faire = fabriquer
Det =: un-Modif, faire = fabriquer
Det =: un, faire = avoir
Det =: Dnum
Det =: un, Advtd

Si cet ensemble de propriétés est accepté, c'est une entrée des tables FD2, FD3A ou FD3B :
Det =: un, faire = pratiquer
Det =: un-Modif, faire = pratiquer

Si aucune des propriétés mentionnées n'est acceptée, l'entrée figure dans la table FD4.

6.4 Les classes des expressions figées

Les classes des expressions figées (cf. Tab. 5.1(c)) sont principalement celles de Gross (1982, 1988, 1996), elles sont au nombre de 54 (en incluant les 4 qui n'ont jamais été publiées). Nous expliquons comment nous avons procédé pour définir leurs constructions de base (6.4.1), comme nous l'avons fait dans Tolone *et al.* (2010). De même, celles de Giry-Schneider (1987) sont détaillées (6.4.2). À la suite de quoi nous nous penchons sur le cas de la classe de verbes 31I qui auparavant n'était pas considérée comme figée (6.4.3), et nous terminons par les constructions en être Prép X étudiées par Danlos (1980) (6.4.4).

6.4.1 Classes d'expressions figées de M. Gross

Chaque classe correspond à une classe morpho-syntaxique des expressions figées (Gross, 1982), le découpage a donc été fait en fonction de la structure de l'expression figée. Ensuite, les intitulés des différentes colonnes qui codent les mots de l'entrée figée correspondent à cette structure.

Nous avons défini chaque construction de base en incluant cette structure dans la phrase, c'est-à-dire en ajoutant le verbe et les arguments libres qui ne sont pas inclus dans l'expression figée[19]. Nous avons gardé la notation N0, N1 et N2 pour les arguments libres, et nous avons utilisé la notation C0, C1 et C2 pour les substantifs figés inclus dans la structure d'un argument figé tel que par exemple, Det1 C1 dans la construction N0 V Prép1 Det1 C1 Prép2 N2.

Cette construction est la construction de base que nous avons défini pour la table CPQ. Cela signifie que le sujet est libre, le verbe est figé, le premier complément est figé (avec pour structure Det1 C1) et introduit par une préposition figée, le deuxième complément est libre et introduit par une préposition figée (par exemple, l'entrée *aboutir à la conclusion de*).

Les tables d'expressions figées avec le verbe *être*, suivi d'une préposition, et un sujet libre (EPAC à EPDETC) sont à traiter à part, car plutôt que de les considérer comme des expressions figées verbales, on peut les considérer comme des expressions figées adjectivales (voire adverbiales) avec le verbe support *être* (*Luc est sans gène* → *Il a fait ça sans gène*). Pour simplifier la classification, nous les avons laissées dans la catégorie des expressions figées.

Si l'on prend par exemple la table EPCPN, la construction de base est N0 être Prép1 Det1 C1 Prép2 N2 (*être à la frontière entre*). Dans cette expression, le second complément libre Prép2 N2 est forcément complexe (*Sartre est à la frontière entre la dépression et le génie*) ou pluriel obligatoire (*Bénazet est à la frontière entre les deux nations*). Le verbe *être* n'est pas figé, il est support car il peut s'effacer (cf. section 3.2) :
Je connais quelqu'un qui est à la frontière entre la dépression et le génie
= *Je connais quelqu'un à la frontière entre la dépression et le génie*

19. Nous avons réalisé ce travail en collaboration avec Stavroula Voyatzi en 2009 (Tolone *et al.*, 2010).

En revanche, la négation éventuelle est figée, car elle ne s'efface pas. C'est pourquoi, nous spécifions le verbe support *être* dans le lexique *LGLex*, comme pour les tables de noms prédicatifs.

Dans chaque table, il y a une colonne pour chaque mot faisant partie de l'expression figée. Toutes les colonnes ont été renommées pour correspondre à la construction de base. Pour la table vu précédemment (EPCPN), on a <ENT>Ppv (qui correspond au codage des colonnes Ppv =: en figé et Ppv =: Neg), <ENT><être>, <ENT>Prép1, <ENT>Det1, <ENT>C1 et <ENT>Prép2. Notons que la colonne <ENT><être> permet de coder le verbe *être*, mais, par exemple, également l'adverbe négatif *pas*. Il est en effet indiqué dans l'entrée verbale, comme pour les tables de verbes, lorsque la colonne <ENT>Ppv est codée *ne* ou *n'*, la colonne Ppv =: Neg est codée +, et que l'adverbe est contraint, ce qui est toujours le cas dans les expressions figées. Notons que les pronoms clitiques et les adverbes figés avec le verbe sont inclus dans l'entrée verbale de la construction de base (c'est-à-dire être, V, etc.), c'est pourquoi Ppv ne figure jamais parmi les constituants de la construction de base. Puis, on a des propriétés distributionnelles sur les arguments libres, telles que N0 =: Nhum, N0 =: N-hum et N0 =: Qu P pour le sujet, ainsi que N2 =: Nhum et N2 =: N-hum pour le second complément libre.

Notons que la notation *<E>* est utilisée pour toute catégorie n'apparaissant pas dans l'entrée, comme par exemple pour l'entrée *être en <E> flirt avec*[20]. Mais il faut surtout préciser que les différents intitulés correspondant aux différentes zones figées des expressions sont approximatifs. En effet, d'une part, les structures sont tellement variées que si l'on voulait donner aux zones des intitulés exacts, cela produirait une multitude de classes minuscules. D'autre part, si l'on supprimait complètement les intitulés, cela gênerait le codage manuel des nouvelles entrées. Ces intitulés approximatifs rendent donc la formalisation exacte très délicate (actuellement, seule l'entrée complète est gardée). En effet, les mots de l'entrée ne correspondent pas toujours à la catégorie syntaxique indiquée par l'intitulé de la colonne, mais ils peuvent appartenir à une autre catégorie ou à une combinaison de plusieurs catégories.

Par exemple, l'entrée *être dans de bonnes dispositions envers* figure également dans la table EPCPN, sans y avoir ajouté une colonne <ENT>Adj1. C'est donc la colonne <ENT>Det1 qui contient le déterminant et l'adjectif *de bonnes*. Ajouter une colonne peut paraître simple, mais il a fallu limiter le nombre de colonnes pour des raisons de lisibilité. Par exemple, la table CPP contient les colonnes <ENT>V, <ENT>Prép1, <ENT>Det1, <ENT>C1, <ENT>Prép2, <ENT>Det2 et <ENT>C2, ce qui permet d'y insérer, sans que cela pose problème, l'entrée *répondre à la force par la force*. Mais cette table contient également l'entrée *vivre de <E> amour et de <E> eau fraîche* où la colonne <ENT>Prép2 contient la conjonction et la préposition *et de*, l'entrée *tomber sur plus fort que <E> Lui-0*, où la colonne <ENT>Prép1 contient l'adverbe *plus* et la colonne <ENT>Prép2 contient la conjonction de subordination *que*, et l'entrée *tirer sur la corde jusqu'à ce*

20. Afin de pouvoir mettre en relation les entrées avec la structure de l'expression figée, nous noterons <E> chaque catégorie vide pour chaque entrée, sauf lorsqu'il s'agit de la colonne <ENT>Ppv puisqu'elle ne figure pas dans la construction de base.

qu'elle casse, où la colonne <ENT>Det2 contient le *ce* de la complétive, tandis que la colonne <ENT>C1 contient la complétive *qu'elle casse*.

Sans compter les expressions figées adjectivales dont nous avons déjà parlé (avec le verbe support *être* et un sujet libre), les autres classes d'expressions figées sont verbales, sauf la table C0E qui rassemble les expressions figées de diverses catégories empoyées en tant que phrases. Elle contient des verbes à l'impératif (*<E> <E> <E>* **sauve** *<E> qui peut ! <E>*), des noms (*<E> au (=à le)*[21] *le* **plaisir** *de vous revoir ! <E>*), des adverbes (*<E> <E> <E>* **prochainement** *sur vos écrans <E>*), des pronoms interrogatifs (*<E> à <E>* **quoi** *<E> <E> bon ? <E>*), etc. Sa construction de base a été définie par Conj Prép0 Det0 C0 Prép1 (Det1 C1+N1) Prép2 N2. Par exemple, pour l'entrée *sauve qui peut !*, le C0 correspond à *sauve*, le Det1 correspond à *qui* et le N1 correspond à *peut !*. Notons que la colonne <ENT>Ppv figure également dans cette table mais est vide pour toutes les entrées sauf une : *ne pas <E> se pencher au dehors <E>*. Elle contient donc le pronom clitique *ne* figé avec le verbe comme dans les autres tables, puisque le C0 correspond au verbe pronominal à l'infinitif *se pencher*[22].

Voyons à présent le cas des trois tables avec le verbe *être* et un sujet figé : EC0, E0P1 et E01. Elles ont les constructions de base suivantes :

- la table EC0 : Ce être Prép1 Det1 C1 Prép2 Det2 C2 (*c'être <E> du beau <E> <E> <E>*) ;
- la table E01 : Det0 C0 être Adj1 (*l'agonie être lente*) ;
- la table E0P1 : Det0 C0 être Prép1 Det1 C1 (*les actions être en baisse* et *les actions être en hausse*).

Pour la table EC0, son sujet figé est *ce* ou *c'*, après avoir codé cette variante dans la colonne <ENT>Ce de la table.

Dans ces tables, la colonne <ENT><être> permet de coder le verbe *être* mais également *pas*, *plus*, *jamais*, *point*, *ni*, *donc*, *bien*, *toujours*, *que* ou une combinaison de ceux-ci. Cela permet aussi d'indiquer dans de rares cas une variante du verbe *être*, telles que *va être*, *Lui être*, *est-ce*, *soit*, *avoir été* ou *es-tu*, dont voici quelques exemples :
<E> ça **va être** *<E> la fête*
<E> il **Lui est** *d' <E> avis que P*
*<E> qu'***est-ce** *qu'il y a ?*
le ciel **soit** *loué !*
la journée **a été** *longue*
Il faut donc traiter ces expressions comme étant des expressions totalement figés, y compris au niveau verbe. A part pour ces cas-là, la colonne <ENT><être> doit être toujours à l'infinitif pour indiquer que le verbe peut être conjugué à tous les modes et à tous les temps. N'ayant aucune information sur la personne, il sera conjugué également

21. Pour plus de lisibilité, les ellisions des déterminants ont été systématiquement faites pour chaque entrée et ne sont indiquées entre parenthèses que dans le cas où la préposition et le déterminant ne forme plus qu'un seul mot, comme c'est le cas ici, pour rendre compte des deux mots codés dans la table.

22. Ppv ne figure pas non plus dans cette table parmi les constituants de la construction de base puisque cela ne concerne qu'une entrée. Aucun <E> ne correspond donc à la colonne <ENT>Ppv.

à toutes les personnes.

De plus, nous avons dupliqué cette colonne pour garder la conjugaison originale telle qu'elle est dans les exemples (colonne Exemple) : la colonne <ENT>est n'est donc là qu'à titre informatif. Dans de rares cas, le verbe est dans un autre temps que le présent :
il était une fois un N qui V W
ça sera un café pour Max !
il fut un temps où P
Ceci était codé à l'origine dans <ENT><être>, nous avons préféré laissé l'infinitif, car dans certains cas tous les temps sont possibles. Par exemple, *c'était donc ça !* ou *ce sera pour une autre fois* (les phrases *c'est pour une autre fois* ou *ce serait pour une autre fois* sont acceptables). Au présent, nous avons conjugué le verbe *être* comme suit :

- *sont* lorsque <ENT>Det0 = les, <ENT>Det0 = Det, suivi d'un nom au pluriel ou <ENT>C0 = ils ;
- *est* lorsque <ENT>Det0 = le, <ENT>Det0 = la, <ENT>Det0 = Det, suivi d'un nom au singulier, <ENT>C0 = il ou <ENT>C0 = on ;
- *suis*, *es*, *sommes* ou *êtes* lorsque <ENT>C0 = je, <ENT>C0 = tu, <ENT>C0 = nous ou <ENT>C0 = vous, ce qui était déjà fait pour certaines entrées.

La construction de base de chaque table dépend donc de la construction prévue au départ par M. Gross, en incluant les intitulés des différentes colonnes qui codent les mots de l'entrée figée. Les constructions de base de chaque table sont détaillées dans la section 6.6. Quelques tables méritent un commentaire, car elles ont plus d'une propriété définitoire :

- les tables C1D, C1DPN, C1I et C1IPN contiennent la nature du déterminant figé, qui peut être soit défini (Det1 =: déf) pour les tables C1D (*abandonner la compétition*) et C1DPN (*abaisser les bras devant*), soit indéfini (Det1 =: indéf) pour les tables C1I (*allumer une mèche*) et C1IPN (*accorder une large confiance à*) ;
- les tables C5 (*bouillonner dans le ciboulot*) et C5C1 (*aller contre l'évidence*) sont des tables à complétives sujet (N0 =: Qu P) ;
- la table E1HC (*être une andouille*) a un sujet humain (N0 =: Nhum) ;
- la table EPAC (*être d'un mauvais acabit*) a pour construction de base N0 être Prép1 Det1 Adj1 C1, l'adjectif étant toujours placé avant le nom. Comme il s'agit de l'ordre inhabituel par rapport aux autres tables, la propriété Adj1 permut obl a été ajoutée pour clarifier l'ordre des mots de l'expression figée ;
- la table CAN (*abattre le moral*) avait à l'origine la construction de base N0 V Det1 C1 (à+de) Nc, que nous avons scindée en deux : N0 V Det1 C1 à Nc comme construction de base et N0 V Det1 C1 de Nc comme deuxième construction définitoire, la table CDN ayant N0 V Det1 C1 de Nc comme construction de base (*allonger le pas*) ;
- la table EC0 (*c'être de l'algèbre <E> <E> <E> pour*) avait à l'origine la construction de base Ce être Prép1 Det1 C1 Prép2 Det2 C2 Prép3 N3 alors que le troisième complément n'est pas toujours possible (*ce n'être pas le tout <E> <E> <E> <E>*, la valeur de la colonne <ENT>Prép3 ne figurant pas dans cette entrée), nous l'avons donc renommée Ce être Prép1 Det1 C1 Prép2 Det2 C2. Ensuite, dans la table,

nous avons ajouté la colonne Ce être Prép1 Det1 C1 Prép2 Det2 C2 Prép3 N3 avec la valeur + lorsqu'au moins l'une des cinq colonnes suivantes vaut + : N3 =: Nhum, N3 =: N-hum, N3 =: ce Qu Pind, N3 =: ce Qu Psubj et N3 =: (de) V-inf W. De plus, la colonne <ENT>Prép3 est mise à − (et non <E>) lorsque cette nouvelle colonne est codée − pour distinguer les cas où la préposition vaut réellement <E> (*c'être le cas de <E> dire <E>*) des cas où le complément n'existe pas ;
- la table CV (*sentir de quel côté venir le vent*) a pour construction de base N0 V Prép1 (Det1 C1+N1) V2 Prépv (Detv Cv+Nv) mais les colonnes Prépv et Detv sont fusionnées en une seule colonne intitulée <ENT>PrépDetv (correspondant à *le* dans l'exemple précédent et à *par* dans *ne demander qu'à se laisser convaincre par*), mais cela ne pose pas de problème pour récupérer l'entrée complète.

6.4.2 Classes d'expressions figées de J. Giry-Schneider

Les tables FC à FCPNN de J. Giry-Schneider sont 9 classes d'expressions figées suivant les mêmes principes que les tables de noms prédicatifs à verbe support *faire* (FN à FNPNN) de J. Giry-Schneider. La différence est que leur déterminant est figé, comme cela est décrit dans Giry-Schneider (1987) (p. 153). Leur construction de base est donc établie de la même manière que pour les classes d'expressions figées précédentes, avec la structure complète dans la phrase.

L'ensemble des tables dont le nom commence par FC ont pour construction de base :
- table FC (*faire l'affaire*) : N0 faire Det1 C1 ;
- table FCA (*faire l'enfant gâté*) : N0 faire Det1 C1 Adj1 ;
- table FCN (*faire la part des choses*) : N0 faire Det1 C1 Prépc Detc Cc.

L'ensemble des tables dont le nom commence par FCPN ont pour construction de base :
- table FCPN (*faire la croix sur*) : N0 faire Det1 C1 Prép2 N2 ;
- table FCPNA (*faire table rase de*) : N0 faire Det1 C1 Adj1 Prép2 N2 ;
- table FCPNN (*faire le pied de grue Loc*) : N0 faire Det1 C1 Prépc Detc Cc Prép2 N2.

L'ensemble des tables dont le nom commence par FCAN ont pour construction de base :
- table FCAN (*faire la gueule à*) : N0 faire Det1 C1 à N2 ;
- table FCANA (*faire la courte échelle*) : N0 faire Det1 C1 Adj1 à N2 ;
- table FCANN (*faire la chasse à l'homme*) : N0 faire Det1 C1 Prépc Detc Cc à N2.

6.4.3 Classe de verbes 31I

La table 31I (sujet Il) est décrite dans Boons *et al.* (1976b) (p. 263), elle est à l'origine une classe de verbes avec comme construction de base Il V (*Il pleut*). Nous la considérons actuellement comme une classe d'expressions figées, car le sujet est figé et noté C0, ce qui est également le cas d'autres classes, comme nous l'avons mentionné précédemment. De plus, des entrées ont été ajoutées n'acceptant pas le sujet *il* mais *ça*, comme dans *Ça dégringole*. Par ailleurs, toutes les entrées acceptant le sujet *il*, acceptent également le sujet *ça* (*Ça pleut*). Nous avons donc noté la construction de base C0 V avec C0 =: ça (et

non N0 =: il+ça, ainsi que cela est noté dans Leclère (1990)), ceci étant également justifié par le fait que la colonne C0 =: il est codée dans la table. Par ailleurs, une construction avec un sujet libre étant également possible pour certains verbes, la colonne N0 V W (*Les tomates pleuvent (E+sur la scène)*) est également codée dans la table.

6.4.4 Constructions en être Prép X

Les tables Z à ZS de L. Danlos décrites dans Danlos (1980) (p. 117) sont au nombre de 8, mais seulement 5 ont été numérisées et corrigées manuellement : Z5D, Z5P, Z1, ZS et ZP [23]. Seules les entrées de ces 5 tables font partie du lexique *LGLex*.

De la même manière que pour certaines tables de M. Gross (EPAC à EPDETC, cf. 6.4.1), plutôt que de les considérer comme des expressions figées verbales, on peut les considérer comme des expressions figées adjectivales (voire adverbiales) avec le verbe support *être*. Le point important est de savoir quels sont le ou les verbes supports possibles, y compris en fonction de l'aspect. Par exemple, pour l'entrée *(être+partir) à la retraite*, *à la retraite* est un adjectif (*Les gens à la retraite sont pauvres*), mais cela n'est pas peut-être pas le cas pour toutes les entrées. C'est pourquoi pour simplifier la classification, nous les avons laissées dans la catégorie des expressions figées. Il faudrait également comparer toutes les entrées avec celles de M. Gross pour voir s'il y a des doublons.

Les propriétés définitoires ont été définies comme suit (la première étant la construction de base) :

- table Z5D (*être à la charge de*) : N0 être Prép1 Det1 C1 de N2, il être Prép1 Det1 C1 de N2 Qu P et N0 =: Qu P. La colonne <ENT>Prép2 ne figurant pas dans cette table, elle a été ajoutée, même si valeur vaut *de* pour toutes les entrées. En effet, cela permet d'avoir l'entrée complète (plus lisible), à savoir avec la préposition figée. Ces modifications ont également été réalisées pour les tables Z1 et ZD ;
- table Z5P (*être de la première urgence pour*) : N0 être Prép1 Det1 C1 (E+Prép2 N2), il être Prép1 Det1 C1 (E+Prép2 N2) Qu P, Prép2 # de et N0 =: Qu P (Z5D et Z5P sont les deux uniques tables à accepter cette propriété). Le deuxième complément n'étant pas toujours possible (*être dans l'ombre d'un doute*, la valeur de la colonne <ENT>Prép2 ne figurant pas dans cette entrée), les deux premières constructions ont été renommées en N0 être Prép1 Det1 C1 et il être Prép1 Det1 C1 Qu P. Ensuite, dans la table, les deux constructions N0 être Prép1 Det1 C1 Prép2 N2 et il être Prép1 Det1 C1 Prép2 N2 Qu P ont été ajoutées avec la valeur − lorsque l'une des 2 colonnes N2 =: Nhum et N2 =: N-hum vaut +. La colonne <ENT>Prép2 contient la préposition codée lexicalement dans ces cas-là également, dans les autres cas, elle vaut −. De plus, la colonne N0 être Prép1 Det1 C1 a été supprimée de la table, puisqu'elle codait la possible omission du N2 et était codée + pour toutes les entrées sauf une, ce qui ne semble pas pertinent ;

23. Nous avons réalisé ce travail en collaboration avec Jean-Pierre Heymann en 2009.

- table Z1 (*être à un poil de*) : N0 être Prép1 Det1 C1 de V0-inf W et N2 =: Qu P (unique table à accepter cette propriété) ;
- table ZS (*être au (=à le) téléphone avec*) : N0 être Prép1 Det1 C1 avec N2 que nous avons renommée N0 être Prép1 Det1 C1 Prép2 N2, la colonne <ENT>Prép2 codant lexicalement la préposition, et N0 et N2 être Prép1 Det1 C1 ;
- table ZD (*être de l'acabit de*) : N0 être Prép1 Det1 C1 de N2 ;
- table ZP (*être aux (=à les) petits soins avec*) : N0 être Prép1 Det1 C1 Prép2 N2 et Prép2 # de ;
- table ZM (*être à l'agonie*) : N0 être Prép1 Det1 C1 et Vop = mettre (propriété qui n'est pas acceptée par la table suivante) ;
- table Z (*être à l'abandon*) : N0 être Prép1 Det1 C1.

Les tables de L. Danlos décrites dans Danlos (1980) ont été reprises par L. Danlos et existent dans une version plus complète (avec plus d'entrées), décrite dans Danlos (1988). Leur nombre est de 24 classes, la plupart ayant été numérisées [24], sans avoir encore été corrigées afin de pouvoir être intégrées (leurs entrées ne font donc pas partie du lexique *LGLex*). Malheureusement, 5 d'entre elles n'ont pas été retrouvées. C'est la raison pour laquelle nous ne faisons pas figurer ces tables dans la section 6.6, nous les détaillons uniquement ici. Nous indiquons pour chaque table de Danlos (1980), les tables en lesquelles elles ont été divisées Danlos (1988) (p. 35) :

- table Z5D : tables Z5CDN (*être du (=de le) goût de*) et ZT5CDN (*être dans les cordes de*) ;
- table Z5P : tables Z5CPN (*être sans <E> espoir pour*) et ZT5CPN (*être à <E> propos*) (qui n'a pas été retrouvée mais qui doit comporter un petit nombre d'entrées que l'on peut retrouver en comparant les tables Z5P et Z5CPN) ;
- table Z1 : tables Z1C (*être dans l'impossibilité de*) et ZT1C (*être à la veille de*) (qui n'a pas été retrouvée et qui ne figure même pas dans l'index, mais on peut retrouver facilement les entrées) ;
- table ZS : tables ZCSY (*être en <E> froid avec*) et ZTCSY (*être en <E> conflit avec*) ;
- table ZP : tables ZCPN (*être en route pour*) et ZTCPN (*être en <E> extase devant*)
- table ZD : tables ZCDN (*être à la merci de*) et ZTCDN (*être sous la botte de*) (on peut donc remplacer ZD par ces tables, sans la numériser) ;
- table ZM : tables ZMC (*être aux (=à les) abois*), ZMAC (*être de <E> bon poil*), ZMCA (*être à l'heure juste*) et ZMCDC (*être aux (=à les) arrêts de rigueur*) (on peut donc remplacer ZM par ces tables, sans la numériser) ;
- table Z : tables ZC (*être de <E> parole*), ZAC (*être de <E> bas étage*), ZCA (*être à <E> ciel ouvert*), ZCDC (*être à la noix de coco*) et ZTCDC (*être à l'article de la mort*), tables que l'on a retrouvées mais aussi ZTC (*être en <E> arrêt*), ZTAC (*être à la dernière extrémité*), ZTCA (*être en <E> terrain acquis*), qui n'ont pas été retrouvées (on ne peut donc pas remplacer Z par ces tables, il faut donc la numériser). Ceci nous empêche de remplacer toutes les tables de Danlos (1980) par

24. Nous avons réalisé ce travail en collaboration avec Jean-Pierre Heymann en 2010.

celles de Danlos (1988).

Ces tables se différencient par le fait que toutes les tables dont le nom commence par ZT ont des variantes aspectuelles utilisant des verbes de mouvement, ce qui implique que le verbe *être* admet des variantes aspectuelles des verbes de mouvement (Vmt) comme *entrer*, *arriver*, *tomber*, ou que le verbe *être* admet des opérateurs causatifs de mouvement (Vcmt) comme *conduire*, *envoyer*, *pousser*, ce qui est noté par les deux constructions suivantes qui sont définitoires de ces tables :
N0 Vmt Prép X W
N Vcmt N0 Prép X W

6.5 Les classes des adverbes

Les classes des adverbes (cf. Tab. 5.1(d)) sont au nombre de 32. Nous commençons par étudier la différences des propriétés présentes dans les différentes tables d'adverbes (6.5.1), puis nous verrons en détail les différentes classes d'adverbes en *-ment* (6.5.2) et d'adverbes (semi-)figés (6.5.2), tout en précisant pour chacune leur construction de base. L'adverbe, qu'il soit simple ou composé, est noté Adv dans les constructions.

6.5.1 Propriétés distinctes

On dispose de deux ressources d'adverbes qui ne traitent pas le même ensemble de propriétés, comme nous l'avons évoqué dans Tolone *et al.* (2010). Il s'agit, d'une part, des adverbes monolexicaux en *-ment* de Moliner (1984); Molinier et Levrier (2000), qui sont dérivés essentiellement des adjectifs et, d'autre part, des adverbes polylexicaux ou complexes (semi-)figés de Gross (1986a, 1990). Ces deux ressources suivent, comme toutes les tables du Lexique-Grammaire, les mêmes principes de représentation : prise en compte de la phrase complète, utilisation d'intitulés compacts et lisibles, etc. C'est pourquoi nous avons jugé intéressant de les combiner, créant ainsi une seule table des classes correspondant à l'ensemble des classes d'adverbes.

Les classes d'adverbes en *-ment* suivent la classification sémantique de la grammaire traditionnelle (temps, manière, etc.) et les propriétés décrivent principalement la place de l'adverbe dans la phrase.

Pour ce qui est des adverbes (semi-)figés, les propriétés décrivent une structure de phrase simple, le plus souvent à prédicat verbal intransitif dont le sujet humain (N0 =: Nhum) ou non humain (N0 =: N-hum), est décrit et codé dans les deux premières colonnes des tables. La représentation de la combinatoire de l'adverbe avec une structure explicite de phrase élémentaire permet de rendre compte des relations de *portée* de l'adverbe sur un élément de la phrase :
La réunion devra avoir lieu ***au plus tard*** *le 15 juin* (table PAC)
Dans l'exemple ci-dessus, l'adverbe *au plus tard* porte sur l'adverbe de date *le 15 juin*. De manière générale, cet adverbe modifie obligatoirement un complément de temps. Des contraintes de temps ou d'aspect s'observent aussi comme dans l'exemple :
Les tablettes remplaceront les PC ***dans un avenir proche*** (table PCA)

** Les tablettes (ont remplacé + remplacent + remplaçaient) les PC* ***dans un avenir proche***

Les propriétés de compléments libres (par exemple, *sous réserve de N*), qui sont importantes pour l'analyse syntaxique, sont codées dans les classes d'adverbes (semi-)figés mais ne figurent pas dans les adverbes en *-ment* (par exemple, *contrairement à N*).

Malgré leurs différences, les deux types d'adverbes sont complémentaires et sont souvent liés par des relations de paraphrase (productives et régulières) permettant de former des couples de synonymes, comme en témoignent les exemples suivants :
pratiquement (table ADVPS) = *en pratique* (table PC)
franchement (table ADVPS) = *à franchement parler* (table PV)
sincèrement (table ADVMS) = *de (manière+façon) sincère* (table PCA)
politiquement (table ADVMP) = *d'un point de vue politique* (table PCA)
malheureusement (table ADVPAE) = *par malheur* (table PC)
Notons qu'il existe au moins deux emplois adverbiaux correspondant à la forme *franchement* : d'une part, adverbe de manière orienté vers le sujet appartenant à la table ADVMS, et qui peut être paraphrasé par *de manière franche* (PCA) et *avec franchise* (PC) ; et, d'autre part, adverbe de phrase disjonctif de style (ou d'énonciation) appartenant à la classe ADVPS et admettant les paraphrases *à franchement parler* (PV) et *en toute franchise* (PDETC). De manière similaire, *sincèrement* a un premier emploi en tant qu'adverbe de manière orienté vers le sujet (paraphrasable par *de (manière+façon) sincère* et *avec sincérité*) et un deuxième emploi en tant qu'adverbe de phrase disjonctif de style (ou d'énonciation) (paraphrasable par *en toute sincérité*). Enfin, *pratiquement* peut être adverbe de manière verbal (ADVMV) ou adverbe de manière focalisateur (ADVMF) ou encore, adverbe de phrase disjonctif de style (ou d'énonciation), paraphrasable par *en pratique* (PC).

Toutefois, cette information significative n'apparaît pas dans les tables du Lexique-Grammaire à cause de la dispersion des adverbes concernés dans les différentes classes. Comme le signale Gross (1990) (p. 56), «seul un système de renvois explicites permettrait ces regroupements sémantiques des adverbes». Il faudrait donc faire référence à l'identifiant de l'adverbe dans les tables d'adverbes simples pour chaque adverbe (semi-)figé concerné et vice-versa.

L'objectif de notre travail, entre autres, est de fournir une description complète à la fois des adverbes en *-ment* et des adverbes (semi-)figés, permettant ainsi d'établir un continuum entre les deux types d'adverbes. C'est pourquoi, à la différence des autres catégories, l'homogénéisation des tables a permis définir la représentation des adverbes [25].

Les principales modifications effectuées concernent : la définition et le codage des constructions de base pour toutes les classes, l'ajout des propriétés lexicales mettant en jeu les noms et les adjectifs qui participent aux procédures transformationnelles et

25. Nous avons réalisé ce travail en collaboration avec Stavroula Voyatzi en 2010-2011 (Tolone *et al.*, 2010). Notons que le même travail combinant les tables de verbes simples et les expressions figées verbales pourrait être envisagé, de même qu'avec les adjectifs quand les tables seront terminées, cependant toutes les raisons énumérées pour les adverbes ne sont pas forcément valables pour les autres catégories.

paraphrastiques, et l'ajout des propriétés de paraphrase [26].

6.5.2 Adverbes en -ment

Les classes d'adverbes monolexicaux en *-ment* de C. Molinier sont divisées en 9 grandes catégories, avec un découpage plus fin pour certaines catégories, ce qui aboutit à un total de 16 classes, comme cela a été présenté dans Sagot et Fort (2007). Une première partition est établie entre adverbes de phrase ou adverbes de statut périphérique, et adverbes intégrés à la proposition, c'est-à-dire rattachés au verbe ou à tout autre constituant de la proposition.

Les adverbes de phrase sont regroupés dans six classes, dont le nom commence par ADVP. On peut distinguer trois grandes classes d'adverbes de phrase :

- les conjonctifs (table ADVPC) nécessitent un contexte gauche auquel ils renvoient (*Hâte-toi,* **autrement** *tu seras en retard*) ;
- les disjonctifs regroupent les autres adverbes de phrase. Parmi ceux-ci, les disjonctifs de style ou d'énonciation (table ADVPS) qualifient la manière de présenter les choses par le locuteur (**Honnêtement***, c'est un fiasco*) ;
- les autres adverbes disjonctifs sont les disjonctifs d'attitude ou d'énoncé, qui se subdivisent en quatre sous-classes : les disjonctifs d'attitude évaluatifs (table ADVPAE : *curieusement*), les disjonctifs d'attitude d'habitude (table ADVPAH : *habituellement*), les disjonctifs d'attitude modaux (table ADVPAM : *certainement*) et les disjonctifs d'attitude orientés vers le sujet (table ADVPAS : *sottement*).

Les adverbes intégrés à la proposition sont organisés en dix classes, dont le nom commence par ADVM. On peut distinguer six grandes classes d'adverbes intégrés à la proposition :

- les adverbes de manière orientés vers le sujet (table ADVMS : *Max regarde* **anxieusement** *l'horizon*) ;
- les adverbes de manière verbaux (table ADVMS : *Max regarde* **fixement** *l'horizon*) ;
- les adverbes de manière quantifieurs (une classe découpée en trois tables dont le nom commence par ADVMQ : *Max aime* **énormément** *ce tableau*). Les adverbes indiquent la complétude (table ADVMQC : *partiellement*) ou l'extension qualitative (table ADVMSQE : *fondamentalement*), les autres quantifieurs étant intensifs (table ADVMQI : *énormément*) ;
- les adverbes de point de vue (table ADVMP : **Légalement***, je suis responsable*) ;
- les adverbes de temps (une classe découpée en trois tables dont le nom commence par ADVMT : *Je suis venue ici* **dernièrement**). On distingue dans cette classe les adverbes de durée (table ADVMTD : *définitivement*), les adverbes de fréquence (table ADVMTF : *constamment*) et les autres adverbes de temps (table ADVMTQ : *dernièrement*).
- les adverbes focalisateurs (ADVMF : *Je lis* **principalement** *des romans*).

26. Signalons que le codage des paraphrases (notamment, le codage des propriétés lexicales associées) n'étant pas terminé, uniquement certaines propriétés de paraphrases apparaissent dans la version 3.3, les autres seront incluses dans la suivante (Tolone et Voyatzi, 2011).

Les classes des adverbes monolexicaux en *-ment* de C. Molinier sont syntaxiquement homogènes, c'est-à-dire que chaque classe correspond à une seule classe syntaxico-sémantique des adverbes. Il a donc été facile d'attribuer les constructions définitoires à chaque classe. Les deux constructions Adv, N0 V W et Adv, N0 ne V pas W étaient initialement codées dans les tables des adverbes de manière et des adverbes de phrase. Nous avons supprimé la colonne lorsque la valeur était constante pour toutes les entrées d'une table, afin de rendre compte des constructions définitoires.

Les classes des adverbes de manière (dont le nom commence par ADVM) ont pour construction de base N0 V Adv W, ce qui signifie que l'adverbe peut être placé après le verbe :
Ce livre est en vente exclusivement sur ce site (table ADVMF)
**Exclusivement, ce livre est en vente sur ce site*
De plus, les tables ADVMP, ADVMS et ADVMTF admettent Adv, N0 V W (l'adverbe peut aussi être placé en tête d'une phrase affirmative) :
Ce livre est en vente régulièrement sur ce site (table ADVMTF)
Régulièrement, ce livre est en vente sur ce site
**Régulièrement, ce livre n'est pas en vente sur ce site*
La table ADVMP admet également Adv, N0 ne V pas W (l'adverbe peut aussi être placé en tête d'une phrase négative) :
Ce concert est musicalement une réussite
Musicalement, ce concert est une réussite
Musicalement, ce concert n'est pas une réussite

Les classes des adverbes de phrases (dont le nom commence par ADVP) ont pour constructions définitoires Adv, N0 V W et Adv, N0 ne V pas W (l'adverbe peut être placé en tête de phrase affirmative ou négative) :
Cinquièmement, Sunrider utilise des sous-produits animaux (table ADVPC)
Cinquièmement, Sunrider n'utilise aucun sous-produit animal
**Sunrider n'utilise cinquièmement aucun sous-produit animal*
De plus, la table ADVPC admet la construction définitoire P1 Adv P2 (l'adverbe peut relier deux phrases).

6.5.3 Adverbes (semi-)figés

Les adverbes (semi-)figés de M. Gross sont polylexicaux ou complexes, c'est-à-dire composés de plusieurs mots. Ils sont regroupés dans des classes en fonction de leur structure morpho-syntaxique, de la même manière que les expressions figées. On a donc diverses colonnes codant la forme des adverbes et leurs variantes éventuelles (effacement de l'adjectif figé, etc.). Ces propriétés de structure n'apparaissent pas dans les tables d'adverbes en *-ment* puisqu'ils ne sont composés que d'un seul mot.

Les classes des adverbes (semi-)figés de M. Gross suivent le découpage suivant :
– table PAC : à modifieur adjectival anteposé ;
– table PADV : monolexical figé ;
– table PC : sans déterminant et sans modifieur ;
– table PCA : à modifieur adjectival postposé ;

– table PCDC : à complément de nom figé de type de C ;
– table PCDN : à complément de nom libre de type de N ;
– table PCONJ : conjonction de deux ou plusieurs adverbes figés ;
– table PCPC : à complément de nom figé de type Prép C ;
– table PCPN : à complément de nom libre de type Prép N ;
– table PDETC : avec déterminant et sans modifieur ;
– table PECO : comparatif rattaché à un adjectif ;
– table PF : phrastique à sujet explicite ;
– table PJC : conjonctif figé ;
– table PPCO : comparatif de structure prépositionnelle ;
– table PV : phrastique à sujet non explicite ;
– table PVCO : comparatif rattaché à un verbe.

Ce découpage ne suit donc pas celui des classes des adverbes en *-ment*. Par exemple, dans la table PAC sont codés à la fois des adverbes conjonctifs (*dans un premier temps*, *en dernier lieu*), des adverbes de phrase (*à Poss0 humble avis*, *en toute première approximation*), des adverbes de temps (*en plein automne*, *depuis cent sept ans*), des adverbes de manière verbaux (*tout bêtement*, *n'importe comment*), etc. C'est pourquoi il n'est pas possible d'établir une construction de base pour une table donnée. Il faudrait en effet coder les constructions pour chaque entrée adverbiale, ce qui serait long à réaliser. Nous avons donc décidé de considérer qu'un adverbe (semi-)figé pouvait se placer n'importe où dans la phrase, même s'il s'agit d'une généralisation abusive. Ainsi, tous les adverbes de la table PC (cf. Fig. 3.6) n'admettent pas Adv, N0 V W, par exemple :
Luc fait du piano à volonté
**À volonté, Luc fait du piano*

Les tables des adverbes (semi-)figés, au nombre de 16 (dont le nom commence par P), admettent donc les trois constructions définitoires suivantes :
N0 V Adv W
Adv, N0 V W
Adv, N0 ne V pas W
De plus, nous avons défini pour chaque classe la construction morpho-syntaxique interne de l'adverbe, appelée *structure de base*. Elle remplace la construction de base n'existant pas ici puisque cette structure de base n'entre pas dans une seule construction, comme c'est le cas pour les expressions figées. En revanche, de même qu'une construction de base, elle peut avoir des variantes, par effacement ou ajout de compléments.

Par exemple, pour la table PCPC, la structure de base est Prép1 Det1 C1 Prép2 Det2 C2, comme c'est le cas pour l'entrée *[changer] du (=de le) jour au (=à le) lendemain*[27]. La colonne Prép1 Det1 C1 est codée dans la table et représente l'effacement du deuxième complément. Cette propriété n'est pas acceptée pour l'entrée précédente, en revanche, elle l'est pour l'entrée *[n'agir] pour <E> rien au (=à le) monde*, et permet de reconnaître la variante *[n'agir] pour <E> rien*.

27. Le verbe entre crochets est codé dans la colonne Prédicat type, il ne fait pas partie de l'entrée, mais est donné à titre indicatif.

De même, que pour la construction de base des expressions figées, chaque structure de base dépend donc des intitulés des différentes colonnes qui codent les mots de l'entrée (semi-)figée. Les détails de chaque structure de base pour chaque table sont dans la section 6.6. Notons que la table PV n'a pas de structure de base bien définie, mais le choix entre les deux structures de base Prép V Prépv Detv Cv et Prép Prépv Detv Cv V, qui sont des colonnes codées dans la table. La structure de base des adverbes simples en *-ment* a simplement été notée Adv.

Nous devons noter la présence d'adverbes libres avec un prédicat d'une autre catégorie pour les trois tables suivantes : PECO contenant des adjectifs prédicatifs composés et PPCO et PVCO contenant des expressions figées verbales. En effet, la table PECO admet également comme construction N0 être Adj, où l'adjectif a une structure en Adj comme Det C (*Max est doux comme un ange*)[28]. Dans cet exemple, *doux comme un ange* est l'adjectif composé, alors que l'adverbe libre a uniquement la structure comme Det C (ici, *comme un ange*). La table PPCO a comme construction N0 V comme Prép Det C (*marcher comme sur des roulettes*) et la table PVCO a comme construction N0 V comme Det C (*partir comme une fusée*)[29]. Dans ces exemples, l'adverbe libre est *comme sur des roulettes* et *comme une fusée*, et possède uniquement la structure comme Prép Det C pour la table PPCO, ou comme Det C pour la table PVCO.

Actuellement, ces adverbes font partie du lexique des adverbes, c'est pourquoi nous appelons les entrées des adjectifs prédicatifs et expressions figées correspondantes des entrées supplémentaires, parallèlement aux autres entrées supplémentaires déjà mentionnées (cf. 6.3.1 et 6.3.3). Lorsque les tables des adjectifs prédicatifs et le lexique qui en découle sera créé, on pourra inclure ces entrées comme des entrées principales dans le lexique des adjectifs prédicatifs ou celui des expressions figées, et supprimer ainsi les entrées adverbiales.

6.6 Recensement

Nous donnons d'abord le nombre d'entrées dans chaque table, puis nous récapitulons l'ensemble des propriétés définitoires détaillées tout au long de ce chapitre, avant de conclure.

6.6.1 Nombre d'entrées de chaque table

Voici le bilan du nombre d'entrées par catégorie (en comptant les expressions figées à part) :

- 13 867 entrées verbales, dont 5 738 verbes distributionnels simples distincts (réparties dans 67 tables) ;
- 12 696 entrées nominales, dont 8 531 noms prédicatifs simples et composés distincts (réparties dans 78 tables) ;

28. L'adjectif *doux* est contenu dans la colonne <OPT>Adj figé.
29. Le verbe *marcher* (respectivement, *partir*) est contenu dans la colonne <OPT>V figé.

– 39 628 entrées figées, dont 38 658 expressions figées distinctes incluant principalement les catégories verbales et adjectivales (réparties dans 69 tables) ;
– 10 488 entrées adverbiales, dont 9 326 adverbes simples et (semi-)figées distincts (réparties dans 32 tables).

Pour plus de détails, le nombre d'entrées par table figure dans les Tab. 6.1 à Tab. 6.7.

Les propriétés codées pour toutes ces entrées sont regroupées dans chacune des tables des classes de la catégorie concernée :
– la table des classes des verbes regroupe 551 propriétés ;
– celle des noms prédicatifs 496 propriétés ;
– celle des expressions figées 276 propriétés ;
– celle des adverbes 159 propriétés.

À cela, on peut ajouter les quelques entrées supplémentaires déjà mentionnées :
– les tables DR1, DRC, FR1, FS1, IS1, ES et AD de G. Gross (cf. 6.3.3) et les tables F1A à F91 (cf. 6.3.2) de J. Giry-Schneider qui codent également des verbes distributionnels simples : elles comportent 3 334 entrées pour 2 377 lemmes distincts (dans 24 tables au total). Remarquons, comme cela a été explicité dans 6.3.2, que les classes de J. Giry-Schneider ne concernent pas de nouveaux verbes, puisque les classes faisant référence aux verbes déjà existants sont indiquées dans la table. De plus, les classes de G. Gross contiennent aussi des verbes déjà existants, mais il est nécessaire les étudier au cas par cas ;
– les tables AN01 à AN06 et ANSY de A. Meunier (cf. 6.3.1), ainsi que la classe ES de G. Gross (cf. 6.3.3) qui codent également des adjectifs prédicatifs simples : elles comportent 1 408 entrées pour 1 287 lemmes distincts (dans 8 tables au total) ;
– les 2 tables d'adverbes qui codent également des expressions figées verbales (PPCO et PVCO) (cf. 6.5.3) : elles comportent 388 entrées pour 385 lemmes distincts ;
– la table d'adverbes PECO (cf. 6.5.3) qui code également des adjectifs prédicatifs composés : elle comporte 304 entrées pour 300 lemmes distincts.

6.6.2 Propriétés définitoires de chaque table

Les Tab. 6.1 à Tab. 6.7 recensent l'ensemble des propriétés définitoires vraies pour toutes les entrées d'une table, qui sont mentionnées dans ce chapitre, incluant les constructions de base ou les structures de base pour les adverbes. De plus, le nombre d'entrées par table figure dans la deuxième colonne. La notation V_ indique qu'il s'agit de classes de verbes, N_ de noms, C_ d'expressions figées et P_ d'adverbes. Nous avons également laissé la notation en minuscules pour les classes de noms, d'expressions figées et d'adverbes, pour des raisons de conformité avec le nom réel des fichiers. Rappelons que pour les verbes, la formule définitoire complète pour chaque table, c'est-à-dire incluant aussi bien les propriétés définitoires vraies que fausses pour l'ensemble d'une table, ainsi que les disjonctions de propriétés codées dans les tables, est donnée dans l'annexe F. Nous avons noté entre accolades les propriétés non présentes dans cette annexe car elles ne participent pas à la définition de la table, mais que nous avons mentionnées dans ce chapitre comme étant constantes +.

Tables	Entr.	Construction de base	Autres propriétés définitoires
V_1	117	N0 V Prép V0-inf W	
V_2	195	N0 V Loc N1 V0-inf W	{N0 V Loc N1}
V_2T	24	N0 V N1 V0-inf W	{N0 V N1}
V_3	54	N0 V N1 Loc N2 V1-inf W	{N0 V N1 V1-inf W ; N0 V N1 Loc N2}
V_4	695	N0 V N1	N0 =: Qu P ; N1 =: Nhum {N0 =: Nnr}
V_5	161	N0 V Prép N1	N0 =: Qu P ; [extrap]
V_6	518	N0 V N1	
V_7	161	N0 V à N1	
V_8	193	N0 V de N1	
V_9	432	N0 V N1 à N2	N2 =: Nhum
V_10	225	N0 V N1 Prép N2	
V_11	226	N0 V N1 à N2	
V_12	62	N0 V N1	N1 =: Qu Psubj ; Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W)
V_13	155	N0 V N1 de N2	
V_14	21	N0 V à N1 Prép N2	
V_15	82	N0 V de N1 Prép N2	
V_16	78	N0 V Prép N1 Prép N2	
V_18	26	N0 V Prép N1 Prép N2 Prép N3	
V_31H	633	N0 V	N0 =: Nhum
V_31R	386	N0 V	N0 =: N-hum
V_32A	162	N0 V N1	N1 apparition ; [passif par]
V_32D	8	N0 V N1	N1 disparition ; [passif par]
V_32C	373	N0 V N1	N1 =: N-hum ; [passif par] {N1 =: Nconc}
V_32CL	121	N0 V N1	N1 =: Npc ; N0 V N1 Loc N1pc W ; [passif par]
V_32CV	114	N0 V N1 en N2	N2 être V-n {N0 V N1 ; N2 apparition}
V_32H	730	N0 V N1	N1 =: Nhum
V_32NM	140	N0 V N1	
V_32PL	139	N0 V N1	N1 =: Npl obl ; [passif par]
V_32R1	98	N0 V N1	N0 V (N1 de N1c) = N0 V (N1c) (Prép N1) ; N1 =: Nabs ; [passif par]
V_32R2	472	N0 V N1	N1 =: Nabs ; [passif par]
V_32R3	1 027	N0 V N1	[passif par]
V_32RA	471	N0 V N1	[passif par] {N0 =: Nnr}
V_33	178	N0 V à N1	
V_34L0	109	N0 V Loc N1	N1 V Prép N0 ; Prép0 =: de
V_35L	94	N0 V Loc N1 source Loc N2 destination	
V_35LD	74	N0 V Loc N1 destination	N1 lieu de destination de N0
V_35LR	169	N0 V Loc N1	N1 = où
V_35LS	24	N0 V Loc N1 source	N1 lieu source de N0 {Loc N1 =: de N1 source}
V_35R	514	N0 V Prép N1	
V_35RR	33	N0 V Prép N1 Prép N2	
V_35S	125	N0 V Prép N1	Prép1 =: avec ; N0 et N1 V
V_35ST	131	N0 V Loc N1	N1 lieu statique de N0
V_39	90	N0 V N1 N2	

TABLE 6.1: Propriétés définitoires (vraies) des tables des verbes distributionnels (1)

Tables	Entr.	Construction de base	Autres propriétés définitoires
V_36DT	275	N0 V N1 Prép N2	{N0 =: Nhum ; N2 =: Nhum ; Prép N2hum = Ppv =: lui}
V_36R	93	N0 V N1 à N2	
V_36S	115	N0 V N1 Prép N2	N0 V N1 et N2
V_36SL	106	N0 V N1 Prép N2	N0 V N1 et N2 {N0 V N1 Loc N2}
V_37E	296	N0 V N1 de N2	{N0 V N1 de N2 = N0 enlever N2 de N1}
V_37M1	220	N0 V N1 Prép N2	Prép2 =: de ; N1 bénéficiaire {N0 V N1 de N2 = N0 mettre N2 Loc N1}
V_37M2	129	N0 V N1 Prép N2	Prép2 =: de ; N1 lieu de destination de N2 {N2 =: trace ; N0 V N1 de N2 = N0 mettre N2 Loc N1}
V_37M3	122	N0 V N1 Prép N2	Prép2 =: de ; N1 lieu de destination de N2 {N2 =: déformation ; N0 V N1 de N2 = N0 mettre N2 Loc N1}
V_37M4	152	N0 V N1 Prép N2	Prép2 =: de ; N1 lieu de destination de N2 {N0 V N1 de N2 = N0 mettre N2 Loc N1}
V_37M5	198	N0 V N1 Prép N2	Prép2 =: de ; N1 lieu de destination de N2 {N2 =: N de V-n ; N0 V N1 de N2 = N0 mettre N2 Loc N1}
V_37M6	283	N0 V N1 Prép N2	Prép2 =: de ; N1 lieu de destination de N2 {N2 =: V-n de N ; N0 V N1 de N2 = N0 mettre N2 Loc N1}
V_38L	123	N0 V N1 Loc N2 source Loc N3 destination	N1 =: N-hum ; N2 lieu source de N1 ; N3 lieu de destination de N1
V_38L0	208	N0 V N1	N1 =: N-hum
V_38L1	164	N0 V N1	N1 =: N-hum
V_38LD	340	N0 V N1 Loc N2 destination	N1 =: N-hum ; N2 lieu de destination de N1
V_38LH	31	N0 V N1 Loc N2 source N3 destination	N2 lieu source de N1 ; N3 lieu de destination de N1 {N1 =: Nhum}
V_38LHD	113	N0 V N1 Loc N2 destination	N2 lieu de destination de N1 {N1 =: Nhum}
V_38LHR	72	N0 V N1 Loc N2	N2 lieu statique de N1 {N1 =: Nhum}
V_38LHS	63	N0 V N1 Loc N2 source	N2 lieu source de N1 {N1 =: Nhum}
V_38LR	284	N0 V N1 Loc N2	N1 =: N-hum ; N2 lieu statique de N1
V_38LS	173	N0 V N1 Loc N2 source	N1 =: N-hum ; N2 lieu source de N1
V_38PL	107	N0 V N1 en N2	N2 =: Npl obl
V_38R	342	N0 V N1 Prép N2	
V_38RR	18	N0 V N1 Prép N2 Prép N3	

Table 6.2: Propriétés définitoires (vraies) des tables des verbes distributionnels (2)

Tables	Entr.	Construction de base	Autres propriétés définitoires
N_aa	64	N0 avoir Det N Prép N1	N1 avoir Det N de N0
N_ad	74	N0 donner Det N à N1	
N_an01	142	N0 avoir Det N	N0 être de Det N Modif ; N0 être Adj
N_an02	154	N0 avoir Det N	N0 être de Det N Modif ; N0 être Adj
N_an03	394	N0 avoir Det N	N0 être de Det N Modif ; N0 être Adj ; Det =: un-Modif
N_an04	157	N0 avoir Det N	N0 être de Det N Modif ; N0 être Adj ; Det =: un-Modif
N_an05	240	N0 avoir Det N	N0 être de Det N Modif ; N0 être Adj
N_an06	276	/	N0 être Adj
N_an07	379	N0 avoir Det N	Det =: un-Modif ; Det =: un-certain
N_an08	617	N0 avoir Det N	Vsup =: comporter ; Vsup =: comprendre
N_an09	270	N0 avoir Det N	
N_an10	489	N0 avoir Det N	
N_andn	60	N0 avoir Det N de N1	
N_ansn	49	N0 avoir Det N sur N1	
N_ansy	24	N0 avoir Det N Prép N1	N0 et N1 avoir Det N ; N0 être de Det N Prép N1 ; N0 être Adj Prép N1 ; N0 et N1 être Adj
N_dr1	234	N0 donner Det N à N1	N1 recevoir Det N (de+de la part de) N0
N_dr2	353	N0 donner Det N à N1	N1 recevoir Det N (de+de la part de) N0
N_dr3	385	N0 donner Det N à N1	N1 recevoir Det N (de+de la part de) N0
N_drc	106	N0 donner un coup de N à N1	N1 recevoir un coup de N (de+de la part de) N0
N_es	90	N0 exercer Det N Prép N1	N1 subir Det N (de+de la part de) N0
N_f1a	30	N0 faire Det N	
N_f1b	57	N0 faire Det N	
N_f1c	96	N0 faire Det N	
N_f1d	3	N0 faire Det N	
N_f1r	44	N0 faire Det N	
N_f21	102	N0 faire Det N Prép N1	Prép1 =: de
N_f2a	379	N0 faire Det N de N1	
N_f2b	40	N0 faire Det N de N1	
N_f2b1	65	N0 faire Det N de N1	
N_f2c	14	N0 faire Det N de N1	
N_f2r	52	N0 faire Det N de N1	
N_f3	83	N0 faire Det N Prép N1	
N_f31	102	N0 faire Det N Prép N1	
N_f4	14	N0 faire Det N à N1	N0 =: Nnr
N_f41	11	N0 faire Det N à N1	N0 =: Nnr
N_f5	237	N0 faire Det N Prép N1	
N_f6	119	N0 faire Det N de N1 Prép N2	
N_f7	52	N0 faire Det N entre N1 et N2	
N_f8	43	N0 faire Det N Prép N1 Prép N2	
N_f9	127	N0 faire Det N de N1	N0 =: Nnr ; Det =: le
N_f91	39	N0 faire Det N de N1	N0 =: Nnr ; Det =: le

Table 6.3: Propriétés définitoires (vraies) des tables des noms prédicatifs (1)

Tables	Entr.	Construction de base	Autres propriétés définitoires
N_fn	373	N0 faire Det N	
N_fna	191	N0 faire Det N	
N_fnan	246	N0 faire Det N à N1	N0 faire Det N
N_fnana	45	N0 faire Det N à N1	N0 faire Det N
N_fnann	90	N0 faire Det N à N1	N0 faire Det N
N_fndn	194	N0 faire Det N de N1	
N_fnn	248	N0 faire Det N	
N_fnpn	414	N0 faire Det N Prép N1	N0 faire Det N
N_fnpna	75	N0 faire Det N Prép N1	N0 faire Det N
N_fnpnn	140	N0 faire Det N Prép N1	N0 faire Det N
N_fr1	111	N0 faire Det N Prép N1	N1 recevoir Det N (de+de la part de) N0
N_fr2	119	N0 faire Det N Prép N1	N1 recevoir Det N (de+de la part de) N0
N_fr3	166	N0 faire Det N Prép N1	N1 recevoir Det N (de+de la part de) N0
N_fs1	1 295	N0 faire Det N Prép N1	N1 subir Det N (de+de la part de) N0
N_fs2	107	N0 faire Det N Prép N1	N1 subir Det N (de+de la part de) N0
N_fs3	91	N0 faire Det N Prép N1	N1 subir Det N (de+de la part de) N0
N_is1	44	N0 infliger Det N à N1	N1 subir Det N (de+de la part de) N0
N_is2	112	N0 infliger Det N à N1	N1 subir Det N (de+de la part de) N0
N_ans	257	N0 Vsup Det N avec N1	N0 et N1 Vsup Det N
N_ana	214	N0 Vsup Det N Prép N1	Prép1 =: à
N_anml	261	N0 Vsup Det N Loc N1pc	
N_anm	121	N0 Vsup Det N	
N_anmr	69	N0 Vsup Det N	
N_ansu	142	N0 Vsup Det N sur N1	
N_anp	64	N0 Vsup Det N pour N1	
N_anpr	52	N0 Vsup Det N Prép N1	
N_ape1	49	N0 avoir Det N	Vsup =: perdre ; Vsup =: conserver ; Vsup =: garder
N_ape11	33	N0 avoir Det N	N0 =: Nhum ; Vsup =: perdre
N_ape2	68	N0 avoir Det N Prép N1	Vsup =: perdre
N_ape21	45	N0 avoir Det N de N1	N1 avoir Det N Prép N0 ; N1 =: Nhum ; Vsup =: perdre ; Vconv =: perdre
N_ape22	56	N0 avoir Det N de N1	N1 Vconv Det N à N0 ; N1 =: Nhum ; Vsup =: perdre ; Vsup =: conserver ; Vsup =: garder
N_ape3	158	N0 avoir Det N Prép N1	N0 avoir Det N ; Vsup =: perdre
N_app1	287	N0 avoir Det N	Vsup =: perdre ; Vsup =: prendre
N_app2	65	N0 avoir Det N Prép N1	Vsup =: perdre ; Vsup =: prendre
N_app3	83	N0 avoir Det N Prép N1	N0 avoir Det N ; Vsup =: perdre ; Vsup =: prendre
N_psy	165	N0 être en N Prép N1	
N_sym	180	N0 être en N Prép N1	N0 et N1 être en N

Table 6.4: Propriétés définitoires (vraies) des tables des noms prédicatifs (2)

Tables	Entr.	Construction de base	Autres propriétés définitoires
C_31i	51	C0 V W	C0 = : ça
C_a1	586	N0 avoir Det1 C1 Prépc Detc Cc	
C_a12	133	N0 avoir Det1 C1 Adj1	
C_a1p2	603	N0 avoir Det1 C1 Prép2 Det2 C2	
C_a1pn	207	N0 avoir Det1 C1 Prép2 N2	
C_anp2	104	N0 avoir N1 Prép2 Det2 C2	
C_c0	1 892	Det0 C0 V Prép1 (Det1 C1+N1) Prép2 N2	
C_c0e	702	Conj Prép0 Det0 C0 Prép1 (Det1 C1+N1) Prép2 N2	
C_c0q	405	Det0 C0 V Prép1 (Det1 C1+N1) Prép2 N2	
C_c1d	1 662	N0 V Det1 C1	Det1 =: déf
C_c1dpn	552	N0 V Det1 C1 Prép2 N2	Det1 =: déf
C_c1g	671	N0 V Poss1 C1	
C_c1gpn	404	N0 V Poss1 C1 Prép2 N2	
C_c1i	1 731	N0 V Det1 C1	Det1 =: indéf
C_c1ipn	938	N0 V Det1 C1 Prép2 N2	Det1 =: indéf
C_c1p2	1 200	N0 V Det1 C1 Prép2 Det2 C2	
C_c1r	589	N0 V Det1 C1	
C_c1rpn	549	N0 V Det1 C1 Prép2 N2	
C_c5	121	N0 V Prép1 Det1 C1 de Nc	N0 =: Qu P
C_c5c1	104	N0 V Prép1 Det1 C1	N0 =: Qu P
C_c6	354	N0 V N1 Prép2 Det2 C2	
C_c7	185	N0 V Det1 C1 à N2	
C_c8	371	N0 V Det1 C1 de N2	
C_cadv	407	N0 V Modif Adv1	
C_can	895	N0 V Det1 C1 à Nc	N0 V Det1 C1 de Nc
C_cdn	855	N0 V Det1 C1 de Nc	
C_cff	206	Conj (Det0 C0+N0) V (Det1 C1+N1) Conj-2 Det0-2 C0-2 V-2 Det1-2 C1-2	
C_cnp2	2 051	N0 V N1 Prép2 Det2 C2	
C_cp1	2 116	N0 V Prép1 Det1 C1	
C_cpn	396	N0 V Prép1 Det1 C1 de Nc	
C_cpp	264	N0 V Prép1 Det1 C1 Prép2 (Det2 C2+N2)	
C_cppn	387	N0 V Prép1 (Det1 C1+N1) Prép2 (Det2 C2+N2) Prép3 (Det3 C3+N3)	
C_cppq	420	N0 V Prép1 (Det1 C1+N1) Prép2 Det2 C2 Prép3 N3	
C_cpq	322	N0 V Prép1 Det1 C1 Prép2 N2	
C_cv	434	N0 V Prép1 (Det1 C1+N1) V2 Prépv (Detv Cv+Nv)	
C_e01	3 043	Det0 C0 être Adj1	
C_e0p1	651	Det0 C0 être Prép1 Det1 C1	
C_e1	280	N0 être Det1 C1	
C_e1hc	457	N0 être Det1 C1 Prépc Detc Cc	N0 =: Nhum
C_e1pn	289	N0 être Det1 C1 Prépc Detc Cc Prép2 N2	
C_eapc	1 304	N0 être Adj1 Prépc Detc Cc	
C_ec0	456	Ce être Prép1 Det1 C1 Prép2 Det2 C2	
C_edn	781	N0 être Det1 C1 de Nc	
C_enpc	630	N0 être Det1 N1 Prépc Detc Cc	

Table 6.5: Propriétés définitoires (vraies) des tables d'expressions figées (1)

Tables	Entr.	Construction de base	Autres propriétés définitoires
C_epa	661	N0 être Adv1 Adj1	
C_epac	743	N0 être Prép1 Det1 Adj1 C1	Adj1 permut obl
C_epc	1 357	N0 être Prép1 C1	
C_epca	710	N0 être Prép1 Det1 C1 Adj1	
C_epcdc	953	N0 être Prép1 Det1 C1 Prépc Detc Cc	
C_epcdn	933	N0 être Prép1 Det1 C1 de Nc	
C_epcpc	902	N0 être Prép1 Det1 C1 Prép2 Det2 C2	
C_epcpn	910	N0 être Prép1 Det1 C1 Prép2 N2	
C_epcpq	185	N0 être Prép1 Det1 C1 Prép2 N2	
C_epdetc	1 060	N0 être Prép1 Det1 C1	
C_fc	215	N0 faire Det1 C1	
C_fca	57	N0 faire Det1 C1 Adj1	
C_fcan	89	N0 faire Det1 C1 à N2	
C_fcana	18	N0 faire Det1 C1 Adj1 à N2	
C_fcann	6	N0 faire Det1 C1 Prépc Detc Cc à N2	
C_fcn	30	N0 faire Det1 C1 Prépc Detc Cc	
C_fcpn	88	N0 faire Det1 C1 Prép2 N2	
C_fcpna	25	N0 faire Det1 C1 Adj1 Prép2 N2	
C_fcpnn	12	N0 faire Det1 C1 Prépc Detc Cc Prép2 N2	
C_ya	351	il y avoir Prép1 Det1 C1 Prép2 (Det2 C2+N2) Prép3 N3	
C_z5d	38	N0 être Prép1 Det1 C1 de N2	N0 =: Qu P ; il être Prép1 Det1 C1 de N2 Qu P
C_z5p	48	N0 être Prép1 Det1 C1	N0 =: Qu P ; Prép2 # de ; il être Prép1 C1 Qu P
C_z1	38	N0 être Prép1 Det1 C1 de V0-inf W	N2 =: Qu P
C_zs	91	N0 être Prép1 Det1 C1 Prép2 N2	N0 et N2 être Prép1 Det1 C1
C_zp	50	N0 être Prép1 Det1 C1 Prép2 N2	Prép2 # de

TABLE 6.6: Propriétés définitoires (vraies) des tables d'expressions figées (2)

Tables	Entr.	Structure de base de l'adverbe	Constructions définitoires
P_advmf	16	Adv	N0 V Adv W
P_advmp	279	Adv	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_advmqc	13	Adv	N0 V Adv W
P_advmqe	21	Adv	N0 V Adv W
P_advmqi	247	Adv	N0 V Adv W
P_advms	883	Adv	N0 V Adv W ; Adv, N0 V W
P_advmtd	11	Adv	N0 V Adv W
P_advmtf	33	Adv	N0 V Adv W ; Adv, N0 V W
P_advmtq	31	Adv	N0 V Adv W
P_advmv	1 523	Adv	N0 V Adv W
P_advpae	10	Adv	Adv, N0 V W ; Adv, N0 ne V pas W
P_advpah	10	Adv	Adv, N0 V W ; Adv, N0 ne V pas W
P_advpam	28	Adv	Adv, N0 V W ; Adv, N0 ne V pas W
P_advpas	30	Adv	Adv, N0 V W ; Adv, N0 ne V pas W
P_advpc	51	Adv	Adv, N0 V W ; Adv, N0 ne V pas W ; P1 Adv P2
P_advps	17	Adv	Adv, N0 V W ; Adv, N0 ne V pas W
P_pac	776	Prép Det Adj C	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_padv	524	Adv	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pc	664	Prép C	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pca	840	Prép Det C Modif pré-adj Adj	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pcdc	750	Prép1 Det1 C1 Prép2 Det2 C2	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pcdn	555	Prép1 Det1 C1 Prép2 N2	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pconj	332	Prép1 Det1 C1 Conj Prép2 Det2 C2	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pcpc	287	Prép1 Det1 C1 Prép2 Det2 C2	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pcpn	151	Prép1 Det1 C1 Prép2 N2	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pdetc	848	Prép Det C	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_peco	304	comme Det C	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pf	396	ConjS (Det0 C0+N0) V Prép1 (Det1 C1+N1)	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pjc	185	ConjC Det C1 Prép C2	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_ppco	50	comme Prép Det C	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pv	285	Prép V Prépv Detv Cv (ou Prép Prépv Detv Cv V)	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W
P_pvco	338	comme Det C	N0 V Adv W ; Adv, N0 V W ; Adv, N0 ne V pas W

Table 6.7: Propriétés définitoires (vraies) des tables d'adverbes

6.6.3 Conclusion

Nous avons vu que les modifications concernant les constructions de base sont diverses :

- certaines propriétés distributionnelles accompagnant la construction de base étaient manquantes ;
- la notation + dans les constructions de base a été au maximum supprimée pour éviter les interprétations ambiguës ;
- certaines colonnes ont été dupliquées pour rendre compte des informations implicites de certaines classes de symétriques ;
- le code des compléments des constructions de base de certaines classes de locatifs a été modifié (par exemple, remplacement de Loc N1 par Loc N1 source et Loc N2 destination), ce qui a impliqué une division des classes ;
- des colonnes ont été ajoutées pour rendre compte de certains cas, comme par exemple les sources dépendantes ;
- de nouvelles classes ont été créées ;
- des colonnes ont été renommées pour être reliées à la construction de base ;
- certaines classes acceptaient à l'origine deux constructions de base mais une seule a été retenue ;
- des colonnes entièrement codées + ont parfois été supprimées pour faire partie des propriétés définitoires ;
- certaines approximations ont été faites lorsqu'il n'existait pas de construction de base ;
- pour les entrées figées (expressions et adverbes (semi-)figés), la structure morpho-syntaxique interne de l'entrée a été définie.

Après avoir défini les constructions de base pour chaque classe dans toutes les catégories (verbes distributionnels, noms prédicatifs, expressions figées et adverbes), cela nous a permis de construire un lexique très riche et d'envisager une utilisation de ces données lexicales dans des outils de traitement automatique, notamment un analyseur syntaxique, comme nous allons le voir par la suite.

Troisième partie

Conversion des tables

7

Conversion au format *LGLex*

Ce chapitre est organisé comme suit. Nous commençons par présenter les motivations de ce travail dans la section 7.1, la version actuelle des tables permettant d'envisager une utilisation de ces données lexicales dans des outils de TAL (Tolone, 2009). Ensuite, nous proposons l'outil générique, nommé *LGExtract*, qui permet de générer un lexique syntaxique pour le TAL à partir des tables du Lexique-Grammaire dans la section 7.2, comme nous l'avons présenté dans Constant et Tolone (2010). Nous expliquons le format du lexique *LGLex* généré pour l'ensemble des catégories, en l'illustrant avec des exemples concrets en français dans la section 7.3, comme nous l'avons initialement fait pour les verbes et les noms (Constant et Tolone, 2010). Puis, nous détaillons le script utilisé pour créer le lexique dans la section 7.4. Enfin, nous discutons des avantages et inconvénients de l'outil *LGExtract* dans la section 7.5.

7.1 Motivations

Un des obstacles à l'utilisation des tables du Lexique-Grammaire en TAL est son degré de formalisation. Il est plus formalisé que le TLFi (Dendien et Pierrel, 2003), dans lequel les propriétés syntaxico-sémantiques sont décrites par du texte ou suggérées par des exemples, et non spécifiées par des intitulés normalisés ; mais il l'est moins que les lexiques utilisés par les analyseurs syntaxiques. Nous présentons les travaux précédents (7.1.1) ayant essayé d'utiliser les tables du Lexique-Grammaire pour l'analyse syntaxique, pour ensuite justifier l'intérêt de notre méthode en introduisant l'outil que nous avons créé, tout en spécifiant sa particularité (7.1.2).

7.1.1 Travaux précédents

Dans les années 2000, les projets LexSynt et LMF (Francopoulo *et al.*, 2006) ont suscité chez les spécialistes de l'analyse syntaxique un renouveau de l'intérêt pour les tables du Lexique-Grammaire. Cela a motivé la recherche de solutions à son insuffisance de formalisation, notamment à travers l'utilisation de réseaux de transitions récursifs (Paumier, 2003; Blanc, 2006). Cependant, le nombre de graphes explose rapidement. Une autre solution envisagée est d'utiliser des formules plus précises que les intitulés, mais elles ne sauraient être aussi concises que ceux-ci (cf. 5.2.1) : la solution n'est donc pas de simplement substituer ces formules aux intitulés, qui gardent leur raison d'être. D'autres (Gardent *et al.*, 2005) préconisent que les informations des tables du Lexique-Grammaire soient rendues utilisables dans des systèmes de traitement des langues par un pré-traitement, qui les ferait passer à un niveau de formalisation équivalent à celui de la norme LMF, et éventuellement soient encodées en XML.

C'est de cette manière que certains travaux (Hathout et Namer, 1998; Gardent *et al.*, 2006; Sagot et Danlos, 2007; Danlos et Sagot, 2008) ont tenté de reformater les tables du Lexique-Grammaire en un lexique syntaxique pour le TAL. Dans ces travaux, à chaque classe est attribué un fichier de configuration spécifique, qui code les informations manquantes et définie les opérations de restructuration. Ainsi, chaque fichier de configuration dans Gardent *et al.* (2006) est représenté par un graphe qui rend la structure de la classe explicite et traduit chaque intitulé de colonne en une structure de traits. Cependant, les tables du Lexique-Grammaire sont continuellement mises à jour pour être améliorées (par exemple, l'ajout ou le renommage de propriétés), ce qui rend cette approche pénible à maintenir. Par exemple, si une même propriété est ajoutée dans plusieurs classes, tous les fichiers de configuration doivent être modifiés.

Cette constatation nous a orienté vers la création de *LGExtract* (Constant et Tolone, 2010) [1], un outil qui utilise une approche globale. D'abord, il est relié à la table des classes, qui code les informations qui ne sont pas définies dans les classes d'origine, particulièrement les propriétés qui sont constantes pour l'ensemble d'une classe. Ensuite, comme une propriété syntaxique a exactement une interprétation pour l'ensemble des classes, notre script d'extraction assigne une seule fois pour chaque propriété un ensemble d'opérations.

Ainsi, nous transcodons les informations des tables du Lexique-Grammaire sous la forme d'ensembles de structures de traits (comparables à celles vu en 5.2.1), dont le format est présenté dans la section 7.3.

Ce qui prime dans cette approche c'est de garder les tables sous leur format tabulaire original pour l'édition manuelle, et ensuite de les convertir dans un lexique syntaxique utilisable en TAL, nommé *LGLex*. Ce traitement permet de relier entre elles les propriétés qui contribuent à décrire une même construction. Par exemple, si l'on prend la construction N0 V à N1 et la propriété distributionnelle N0 =: Nhum : les tables du Lexique-Grammaire elles-mêmes ne les relient pas explicitement, si ce n'est à travers le symbole N0 contenu dans les deux intitulés. Dans cet exemple, on pourrait utiliser l'inti-

1. Disponible sur le site http://infolingu.univ-mlv.fr/ (Données Linguistiques > Lexique-Grammaire > Téléchargement).

tulé N0hum V à N1, comme nous l'avons expliqué en 5.2.2. Mais si l'on souhaite également accepter la propriété distributionnelle N0 =: N-hum, il n'est pas possible de le spécifier dans les tables avec un seul intitulé succinct, or c'est précisément ce que permettent de faire les structures de traits.

7.1.2 Particularité de l'outil *LGExtract*

LGExtract a vocation à transformer les tables en un lexique syntaxique pour le TAL, reposant sur les mêmes concepts linguistiques que ceux qui sont à l'œuvre dans les tables. C'est-à-dire que le format du lexique *LGLex* est ce qu'on appelle un format d'échange. Il a vocation à décrire les tables avec les concepts manipulés par celles-ci, en un format directement exploitable dans les applications de TAL. L'une des utilisations informatiques possibles est la conversion en un autre format, par exemple au format Le*fff*, ce qui suppose de manipuler d'autres concepts linguistiques, ceux manipulés par le Le*fff*.

En effet, contrairement à *LGLex*, qui liste toutes les constructions acceptées par une entrée telles qu'elles existent dans les tables, le Le*fff* regroupe dans une représentation unique des constructions qu'il considère comme étant des variantes l'une de l'autre. Ainsi, l'effacement d'un argument dans une construction est considéré comme une variante de cette construction. Cette représentation est adaptée pour certains formalismes d'analyseurs syntaxiques, tel que TAG utilisé dans l'analyseur FRMG et LFG dans l'analyseur SxLFG (cf. 2.1.3).

Il y a donc deux étapes bien distinctes, puisqu'il ne semble pas souhaitable que le format *LGLex* manipule des concepts issus du Le*fff* ou de DICOVALENCE : le format *LGLex* doit être utilisable par tous les connaisseurs des tables du Lexique-Grammaire, y compris ceux qui refusent totalement la notion de fonction syntaxique (du Le*fff*) ou de paradigme (de DICOVALENCE). La construction du lexique *LGLex* n'a rien à voir avec ses utilisations, comme par exemple sa conversion en un autre format.

LGExtract se concentre sur l'explicitation de certaines colonnes, sans utiliser d'autres concepts que ceux des tables du Lexique-Grammaire. On obtient ainsi dans *LGLex*, une représentation plus explicite des tables, ce qui sert (entre autres) à produire plus simplement une représentation au format Le*fff*.

7.2 L'outil *LGExtract*

Les propositions antérieures pour transformer les tables du Lexique-Grammaire en un lexique syntaxique pour le TAL consistaient en des paramètres spécifiques pour chaque classe : la sélection des propriétés pertinentes, l'ajout de propriétés manquantes et la restructuration de données (Hathout et Namer, 1998; Gardent *et al.*, 2006). Comme plusieurs propriétés apparaissent dans plusieurs classes, la définition d'un même ensemble d'opérations peut ainsi se répéter dans plusieurs classes, ceci rendant cette approche pénible pour l'encodage et la maintenance.

Nous proposons une approche plus globale, en utilisant pour l'ensemble des classes d'une même catégorie les deux ressources suivantes :
- un script unique de configuration couvrant toutes les classes ;
- une table des classes fournissant des informations non définies dans les classes d'origine.

Pour implémenter cette approche, nous avons développé en Java un outil générique nommé *LGExtract* [2], qui fonctionne de la manière suivante :
- il prend en entrée un script de configuration et une table des classes ;
- il analyse ce script à l'aide d'un analyseur généré par l'outil Tatoo (Cervelle *et al.*, 2006) ;
- il produit l'ensemble des entrées lexicales encodées dans les classes couvertes par la table des classes et dans le format décrit par le script.

Les deux instructions principales interprétées par notre outil sont les suivantes :
- **define** : l'information est encodée dans des objets linguistiques définis dans le script. Ils sont représentés par des listes et des structures de traits, qui peuvent être combinées. Ces objets définissent par exemple, des constituants syntaxiques, des distributions de constituants syntaxiques, des constructions, des représentations prédicat-argument, des transformations. Les objets peuvent être paramétrés par les propriétés syntaxiques disponibles dans la table des classes ;
- **prop** : chaque propriété de la table des classes est associée à un ensemble d'opérations qui combinent les objets linguistiques entre eux. Ainsi, quand la propriété N0 =: Nhum est vraie pour une entrée donnée, un objet définissant un groupe nominal humain est ajouté à la distribution de `N0` (c'est-à-dire l'argument `0` du prédicat).

Cela implique que chaque propriété a une et une seule interprétation pour toutes les classes, si tel n'était pas le cas, notre outil fournirait des informations incorrectes.

Un objet linguistique est constitué de listes et de structures de traits. Une instance d'un tel objet est définie par l'instruction **define**, en indiquant son type, son nom et sa valeur. Par exemple, les instructions ci-dessous instancient chacune un composant (**comp**) nommé N-hum, qui est un groupe nominal non humain, un groupe nominal humain (Nhum), une complétive au mode indicatif (Qu Pind), une complétive au mode subjonctif (Qu Psubj) ou une infinitive (V-inf W) :

```
define comp N-hum [cat="NP",nothum="true"];
define comp Nhum [cat="NP",hum="true"];
define comp completive [cat="comp",mood="ind"];
define comp completiveSubj [cat="comp",mood="subj"];
define comp inf [cat="inf"];
```

Ces différents objets peuvent être combinés : par exemple, la distribution d'un constituant (**const**) est un ensemble de composants syntaxiques. Dans l'instruction ci-dessous, le constituant **N0** contient la distribution de l'argument `0` :

2. Il est composé de 118 fichiers *.java* et est disponible sur le site http://infolingu.univ-mlv.fr/ (Données Linguistiques > Lexique-Grammaire > Téléchargement).

```
define const N0 [pos="0",dist=()];
```

avec **dist()** qui pourra contenir un groupe nominal humain (Nhum) et un groupe nominal non humain (N-hum)[3] :

```
comp=[cat="NP",nothum="true"],comp=[cat="NP",hum="true"]
```

Comme dans tout langage orienté objet, un mécanisme d'héritage existe. Par exemple, une infinitive contrôlée par l'argument 0 (objet **inf0**) hérite des traits de l'objet **inf** (définissant une infinitive) décrit ci-dessus, et possède un trait supplémentaire indiquant le contrôle par l'argument 0 :

```
define comp inf0 inf[contr="0"];
```

Tous ces objets peuvent être paramétrés avec les propriétés de la table des classes (notation @...@). Les paramètres sont de deux types : booléen ou chaîne de caractères. Par exemple, le code ci-dessous définit un prédicat verbal nommé **predV**, en l'ajoutant dans les informations lexicales liées à l'entrée (**lexical-info**). Son lemme est la valeur de la propriété <ENT>V (c'est-à-dire la valeur lexicale d'une entrée) dans la classe correspondante. Le code définit ensuite l'auxiliaire *avoir* (respectivement, *être*), où la propriété Aux =: avoir (respectivement, Aux =: être) est encodée par sa valeur booléenne :

```
define lexical-info predV [cat="verb",verb=[lemma="@<ENT>V@"],aux-list=()];
define aux avoir {avoir="@Aux =: avoir@"};
define aux etre {être="@Aux =: être@"};
```

Pour chaque entrée lexicale, les paramètres des objets linguistiques associés sont établis comme suit. Chaque paramètre, correspondant à une propriété, possède une valeur lexicale ou booléenne. Le programme parcourt d'abord la table des classes. Si la propriété a une valeur constante sur toute la classe à laquelle l'entrée appartient, la propriété reçoit cette valeur. Si la valeur de la propriété est variable selon les entrées lexicales (la valeur de la propriété est o pour la ligne correspondant à cette classe), le programme récupère la valeur de la propriété de cette entrée dans la classe concernée. Par exemple, le verbe *alarmer* appartient à la classe 32H, qui contient les verbes transitifs avec sujet humain : **@<ENT>V@** est donc remplacé par la valeur lexicale **alarmer**, ce qui permet de savoir de quel lemme il s'agit pour cette entrée. La propriété Aux =: avoir est codée + dans la table pour ce verbe : **@Aux = : avoir@** est remplacé par **true**. En revanche, la propriété Aux =: être est codée −, donc le programme ne récupère pas cette information (voir plus loin). Les trois objets paramétrés montrés ci-dessus deviennent alors les deux objets suivants pour cette entrée :

```
define lexical-info predV [cat="verb",verb=[lemma="alarmer"],aux-list=()];
define aux avoir {avoir="true"};
```

3. Nous verrons plus loin que cela est obtenu grâce à l'opération d'ajout :
add N0-hum in N0.dist ;
add N0hum in N0.dist ;

Ainsi, si une contradiction survient entre la table des classes et une classe particulière, la priorité est donnée à l'encodage de la table des classes.

Pour chaque entrée lexicale, le programme peut ensuite appliquer des opérations pour chaque propriété de la table des classes à ces objets *lexicalisés*, avec l'instruction **prop**. Il y a seulement un type d'opération : l'ajout (**add**) d'un objet à un autre. Par exemple, l'ajout d'une paire attribut-valeur ou d'une liste dans une structure de traits. Ces opérations sont indépendantes de leur ordre d'application, c'est-à-dire qu'elles sont non destructrices et ne dépendent pas les unes des autres. Ainsi, lors de l'insertion d'une paire attribut-valeur *(a,v)* dans une structure de traits, si une autre valeur *ov* pour l'attribut *a* existe déjà, la nouvelle valeur est une disjonction de *v* et *ov*. C'est pourquoi l'opération est dite *non destructrice*. Les listes sont en fait des ensembles car le résultat des deux additions doit être indépendant de leur ordre d'application. Avant d'insérer un nouvel élément dans une liste, le programme vérifie s'il existe ou non. S'il existe, il n'est pas inséré. Par exemple, le code suivant indique que, si la propriété N0 =: Nnr (signifiant que N0 est un groupe nominal libre, une complétive ou une infinitive) est codée + , le programme ajoute les objets **N-hum**, **Nhum**, **completive**, **completiveSubj** et **inf** à la distribution de **N0** (**N0.dist**) et insère **N0** dans la liste des arguments (**args**) :

```
prop @N0 =: Nnr@{
  add N0 in args;
  add N0-hum in N0.dist;
  add N0hum in N0.dist;
  add completive in N0.dist;
  add completiveSubj in N0.dist;
  add inf in N0.dist;
}
```

Si la propriété est codée − (que ce soit dans la table des classes ou dans la table), aucune opération n'est réalisée. En effet, pour chaque entrée, seules sont effectuées les opérations concernant les propriétés acceptées par l'entrée. C'est ce qui permet dans l'exemple précédent d'ajouter dans le lexique (dans la liste **aux-list** de **predV**) l'objet paramétré **avoir="@Aux = : avoir@"**, sans le faire pour **être="@Aux = : être@"**, avec le code suivant :

```
prop @Aux =: avoir@{
  add avoir in predV.aux-list;
}
prop @Aux =: être@{
  add etre in predV.aux-list;
}
```

Le lexique généré est au format XML. Les éléments et attributs XML peuvent être mis en correspondance dans le script avec les objets linguistiques. Ce lexique XML étant quasiment illisible par un humain, une version texte compressée a été également développée (voir les exemples de la section 7.3).

7.3 Le lexique *LGLex*

Grâce à *LGExtract* et à tous les travaux de mise en cohérence et d'explicitation des propriétés syntaxiques vus aux chapitres 5 et 6, il a été possible de construire un lexique syntaxique du français pour le TAL : *LGLex* (Constant et Tolone, 2010). Il a été généré à partir des tables du Lexique-Grammaire librement distribuées de toutes les catégories, à savoir les verbes distributionnels, les noms prédicatifs, les expressions figées et les adverbes. Il est également librement distribué sous licence LGPL-LR (`http://infolingu.univ-mlv.fr`, Données Linguistiques > Lexique-Grammaire > Téléchargement). Les détails pour pouvoir générer le lexique *LGLex* sont expliqués dans l'annexe B.

Il est composé de 13 867 entrées verbales (provenant de 67 tables), de 12 696 entrées nominales (provenant de 78 tables), de 39 628 expressions figées principalement verbales et adjectivales (provenant de 69 tables) et de 10 488 entrées adverbiales (provenant de 32 tables)[4]. De plus, il contient comme entrées secondaires : 3 334 entrées verbales (provenant de 24 tables de noms prédicatifs), 1 408 entrées adjectivales (provenant de 8 tables de noms prédicatifs), 388 expressions figées verbales (provenant de 2 tables d'adverbes) et 304 entrées adjectivales (provenant d'une table d'adverbes).

Le script d'extraction (présenté dans la section 7.4) encode seulement une sélection de propriétés, certaines ayant été éliminées car elles ne sont pas exploitables. C'est ainsi que nous avons éliminé les propriétés concernant les noms dérivés de verbes (V-n) sans information explicite sur la procédure de dérivation[5]. De même, certaines propriétés concernant des traits sémantiques tels que les noms de parties du corps, le sens métaphorique de la phrase (esprit, idée, mot, texte), etc. n'ont pas été jugées pertinentes. Prenons comme exemple la propriété N2 =: déformation de la table 37M3, indiquant que l'objet N2 peut dénoter le type de déformation subit par l'objet N1, comme dans *La pluie a sillonné le terrain de crevasses*. Il faudrait donc établir une liste précise de tous les noms désignant une déformation, ce qui n'est pas évident car les critères ne sont pas clairement définis. Ainsi, les noms *trou*, *creux*, *cavité*, etc. sont des synonymes de *crevasse* dans l'exemple précédent. Mais si l'on prend l'exemple *Cet accident a couturé son visage de cicatrices*, on doit tenir compte des noms comme *brûlure*, *ride*, *balafre*, etc. qui sont des déformations du corps humain. Enfin, l'exemple *Max a altéré son texte d'annotations* illustre la difficulté d'établir une liste exhaustive, puisque le mot *annotation* est loin de désigner une déformation, ce qui pourtant ce révèle être vrai dans l'exemple précédent. En effet, comme nous l'avons mentionné en 6.2.2, les critères sémantiques ou morphologiques justifiant le découpage des tables 37M2 à 37M6 ne sont pas définissables clairement et n'ont pas la moindre reproductibilité.

Dans le lexique *LGLex*, les informations syntaxiques sont représentées de façon par-

4. Les entrées ne sont autres que celles des tables, le nombre de lemmes distincts est donc également le même (voir la section 6.6).

5. De plus, malgré le codage de certaines informations concernant les V-n dans les tables de verbes (Paumier, 2003), la reconnaissance des V-n est un problème compliqué qui soulève de nombreuses interrogations pour l'instant sans réponse.

tiellement formalisée. Dans sa version textuelle, une entrée de *LGLex* se présente comme suit :

- l'entrée commence par un identifiant indiquant sa catégorie, la table dont il provient et le numéro de l'entrée dans cette table (**ID=catégorie_numTable_numEntrée**), suivi du statut **completed** pour une entrée entièrement codée, **to complete** pour une entrée qui a au moins une propriété non codée [6] ou **to encode** pour une entrée qui a au moins la moitié de ses propriétés non codées [7] ;
- la section **lexical-info** indique les informations lexicales liées à l'entrée :
  - le lemme (correspondant à l'entrée complète, qu'elle soit simple ou composée), et pour les entrées composées, les différents mots de l'entrée associés à leur catégorie grammaticale, ainsi que pour certaines entrées nominales, l'adjectif ou le verbe morphologiquement dérivé du nom ;
  - mais également les auxiliaire(s) pour les entrées verbales, les verbes supports et les déterminants pour les entrées nominales, et les prépositions associées à certains arguments. Les prépositions possibles à la place de l'indication Prép (respectivement, Loc) sont dans la partie **prépositions** (respectivement, **locs**) ;
- la section **args** décrit les distributions des différents arguments (sujet et compléments, répartis en sous-sections **const** dont la position est repérée par l'élément **pos**). Une distribution donnée (élément **comp**) indique :
  - sa catégorie grammaticale : **NP** pour un syntagme nominal, **inf** pour une infinitive (V-inf W), **comp** pour une complétive (Qu P), **leFaitComp** pour le groupe nominal le fait que P, **siPOuSiP** pour la complétive si P ou si P, **adj** pour un adjectif ;
  - son introducteur (élément **introd-prep** ou **introd-loc**) ;
  - des traits sémantiques : **hum**, **nothum**, **plobl** (pluriel obligatoire), **npr** (nom propre), **abst** (abstrait), **conc** (concret), **source**, **destination**, **benef** (bénéficiaire), **mesure**, **prix**, **coll** (collectif), **plur** (pluriel) ;
  - des traits complémentaires (**mood** dans le cas d'une complétive, **contrl** dans le cas d'une infinitive) ;
  - l'intitulé complet des propriétés distributionnelles ayant contribué à définir cette distribution, qui sont toutes de la forme argument =: réalisation. Cette information n'est là qu'à titre informatif, pour pouvoir faire le lien avec les tables d'origine.

  Cette section est donc bien formalisée ;
- la section **all-constructions** liste différentes constructions dans lesquelles l'entrée peut prendre part :
  - les constructions absolues (élément **absolute**) sont les constructions additionnelles à la construction de base, qui sont nommées de façon complète, avec tous les

6. On appelle ici une propriété non codée, une propriété présente dans la table et codée ∼. Cela ne tient pas compte du codage O dans la table des classes, qui signifie également que la propriété doit être codée, mais qui ne figure pas dans la table.

7. Le statut **to complete** permet de différencier ces entrées de celles entièrement codées (statut **complete**), leur codage doit donc être complété. Le statut **to encode** permet de repérer les entrées où seule la table des classes code certaines constantes et non la table concernée elle-même, il faut donc les coder entièrement. En effet, même si une entrée est codée entièrement ∼, elle est difficilement détectable dans le lexique.

éléments dans l'ordre (par exemple, N0 V N1) ;
  - les constructions relatives (élément **relative**) sont les propriétés transformationnelles, qui correspondent aux redistributions (par exemple, [passif par]), pronominalisations (par exemple, Ppv =: y), etc.
  - les réductions en groupe nominal (élément **reductionsGN**) sont pour les entrées nominales, les réductions de la construction de base vers une autre construction dans une autre catégorie syntaxique, dans le cas présent, un groupe nominal (par exemple, le N entre N0 et N1) ;
  - les constructions verbales (élément **verbales**) sont les constructions acceptées par le verbe associé à une entrée nominale ;
  - les structures des adverbes (semi-)figés (élément **structureAdv**) définissent l'ordre des différents constituants d'une entrée adverbiale, ainsi que les possibles variantes (par exemple, Prép1 Det1 C1 Prép2 Det2 C2 et Prép1 Det1 C1, cf. 6.5.3).

  Les constructions codées + (incluant la construction de base) dans la table des classes sont précédées de la mention **true::**, les autres sont précédées de la mention **o::**. Cette section est donc bien moins bien formalisée que la précédente, et les informations qu'elle contient ne sont pas directement exploitables informatiquement ;
- la section **example** illustre l'entrée.

Pour plus de détails, voir l'annexe A.

L'exemple ci-dessous montre un extrait du lexique *LGLex* pour l'entrée verbale *se rendre* de la table 33 des verbes (cf. Fig. 3.1)[8]. La construction de base est N0 V à N1 et l'argument N1 est effaçable puisque l'entrée accepte également la construction N0 V à N1. L'argument N0 doit être un groupe nominal humain, alors que l'argument N1 peut être humain, non humain ou prendre la forme *le fait que P*, proche d'une complétive. Il accepte deux constructions transformationnelles, qui sont [extrap] et Ppv =: y :

```
ID=V_33_130;status=completed
lexical-info=[cat="verb",
              verb=[lemma="rendre",ppvse="true",
              aux-list=(etre="true"),prepositions=(),locatifs=()]]
args=(const=[pos="0",
             dist=(comp=[cat="NP",hum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(orig="N0 =: Nhum")])],
      const=[pos="1",
             dist=(comp=[cat="NP",hum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(orig="N1 =: Nhum")]
                   comp=[cat="NP",nothum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(orig="N1 =: N-hum")]
                   comp=[cat="leFaitComp",
                         introd-prep=(),introd-loc=(),
                         origin=(orig="N1 =: le fait Qu P")])])
```

8. La section 9.3 contient un autre extrait du lexique *LGLex* pour les entrées verbales *croupir*, *croustiller* et *croûter* de la table 31R.

```
all-constructions=[absolute=(construction="true::N0 V à N1",
                             construction="o::N0 V"),
                   relative=(construction="[extrap]",
                             construction="Ppv =: y")]
example=[example="Max s'est rendu à mon(opinion+avis)"]
```

Voici un exemple du lexique *LGLex* pour l'entrée nominale *canular* de la table FNAN des noms prédicatifs (cf. Fig. 3.4). La construction de base est N0 faire Det N à N1 et l'argument N1 est effaçable. Les arguments N0 et N1 doivent être des groupes nominaux humains. Il accepte les deux constructions N0 faire le N de V0-inf W et N0hum faire Det N à N1hum sur ce point :

```
ID=N_fnan_29;status=completed
lexical-info=[cat="noun",
              Vsup=[cat="verb",list=(value="faire")],
              noun=[notperm=[complete="canular"],noun1="canular"],
              detN=[list-det-modif=(det-modif=[det="un+une",modif="false"],
                                    det-modif=[det="un+une",modif="true"],
                                    det-modif=[det="des",modif="false"],
                                    det-modif=[det="<E>",modif="false"])],
              prepositions=()]
args=(const=[pos="0",
             dist=(comp=[cat="NP",hum="true"])],
      const=[pos="1",
             dist=(comp=[cat="NP",hum="true"])])
all-constructions=[absolute=(construction="true::N0 faire Det N à N1",
                             construction="true::N0 faire Det N",
                             construction="o::N0 faire le N de V0-inf W",
                             construction="o::N0hum faire Det N à N1hum
                                                            sur ce point"),
                   relative=(),
                   verbales=(),
                   reductionsGN=()]
example=[example=]
```

Voyons à présent un exemple du lexique *LGLex* pour l'entrée figée *avoir sur le cœur* de la table C6 des expressions figées (cf. Fig. 3.5). Chacun des mots de l'entrée figée est découpé selon sa catégorie (verbe, préposition du deuxième argument Prép2, déterminant du deuxième argument Det2, nom figé C2 faisant également partie du deuxième argument) et le champ **complete** contient l'expression complète. La construction de base est N0 V N1 Prép2 Det2 C2, ce qui signifie que l'argument N1 est libre. L'argument N0 doit être un groupe nominal humain, alors que l'argument N1 peut être humain, non humain, une complétive à l'indicatif ou subjonctif. Il accepte la propriété transformationnelle Qu Pind = Aux V0-inf W, qui signifie que la complétive objet direct à l'indicatif, lorsque son sujet est coréférent au sujet de la principale et qu'elle contient un auxiliaire ou certains verbes (Gross, 1975) (p. 76), peut être remplacée par une infinitive non prépositionnelle dont le sujet implicite est coréférent au sujet de la principale :

```
ID=C_c6_210;status=completed
```

```
lexical-info=[cat="expr",
              exprF=[expr=[notperm=[complete="<E> avoir sur le coeur"],
                           verb="avoir",
                           prep2="sur"
                           det2="le",
                           c2="coeur"]]]

args=(const=[pos="0",
             dist=(comp=[cat="NP",hum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: Nhum)])],
      const=[pos="1",
             dist=(comp=[cat="comp",mood="subj",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: Qu Psubj)],
                   comp=[cat="leFaitComp",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: le fait Qu P)],
                   comp=[cat="NP",nothum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: N-hum)],
                   comp=[cat="NP",hum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: Nhum)],
                   comp=[cat="comp",mood="ind",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: Qu Pind)])])
all-constructions=[absolute=("true::N0 V N1 Prép2 Det2 C2"),
                   relative=(construction="Qu Psubj = de V0-inf W",
                             construction="Qu P = ceci",
                             construction="Qu P = Ppv")]
```

Enfin, voici un exemple du lexique *LGLex* pour l'entrée adverbiale *de visu* de la table PC des adverbes (cf. Fig. 3.6). Comme il s'agit d'un adverbe (semi-)figé, de même que pour les expressions figées, chacun des mots de l'entrée figée est découpé selon sa catégorie (préposition Prép, nom figé C). Sa structure de base est Prép C, accepte les trois constructions définitoires N0 V Adv W, Adv, N0 ne V pas W et Adv, N0 V W, et le sujet doit être humain :

```
ID=P_pc_656;status=completed
lexical-info=[cat="adv",
              exprF=[expr=[notperm=[complete="de visu"],
                           prep="de",
                           c="visu"]]]
args=(const=[pos="0",
             dist=(comp=[cat="NP",hum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(N0 =: Nhum)])],
all-constructions=[structureAdv=(construction="true::Prép C"),
                   absolute=(construction="true::N0 V Adv W",
                             construction="true::Adv, N0 V W",
                             construction="true::Adv, N0 ne V pas W"),
```

```
                    relative=()]
```

Un travail en cours est d'inclure les propriétés de paraphrases, comme vu en 6.5.1 (à Adv parler, P ou N0 V W de (façon+manière) Adj), en tant que variantes des entrées adverbiales dans le lexique *LGLex*, mais également les propriétés de structures, comme vu en 6.5.3 (Prép1 Det1 C1, la structure de base étant Prép1 Det1 C1 Prép2 Det2 C2), et les propriétés d'intensification (plus Adv).

Nous projetons donc d'ajouter les champs suivants dans **lexical-info** contenant ces variantes :

- **paraphrases** (par exemple, *à franchement parler* pour l'adverbe *franchement* et *de (manière+façon) sincère* pour l'adverbe *sincèrement*) ;
- **autres-structures** (par exemple, *jusqu'à la fin* pour l'adverbe *jusqu'à la fin des temps*) ;
- **autres-ID**, désignant les autres entrées avec intensification (par exemple, *plus particulièrement* pour l'adverbe *particulièrement*).

L'exemple suivant montre un exemple du lexique *LGLex*[9] pour l'entrée adverbiale *jusqu'à la fin des (=de les) temps* de la table PCDC des adverbes. On peut y voir le nouvel adverbe *jusqu'à la fin*, qui est ajouté dans **lexical-info** en tant qu'autre structure, suivi par sa structure morpho-syntaxique interne Prép1 Det1 C1, qui est ajoutée dans **all-constructions** :

```
ID=P_pcdc_270;status=completed
lexical-info=[cat="adv",
              exprF=[adv=[notperm=[complete="jusqu'à la fin de les temps"],
                     prep1="jusqu'à",
                     det1="la",
                     c1="fin",
                     prep2="de",
                     det2="les",
                     c2="temps"]]]
              paraphrases=(),
              autres-structures=(adv="jusqu'à la fin"]),
              autres-ID=()]
args=(const=[pos="0",
             dist=(comp=[cat="NP",nothum="true",
             introd-prep=(),introd-loc=(),
             origin=(N0 =: N-hum)])])
all-constructions=[structureAdv=(construction="true::Prép1 Det1 C1 Prép2 Det2 C2",
                                 construction="o::Prép1 Det1 C1"),
                   absolute=(construction="true::N0 V Adv W",
                             construction="true::Adv, N0 V W",
                             construction="true::Adv, N0 ne V pas W"),
                   relative=()]
```

9. Signalons que le codage des paraphrases (notamment, le codage des propriétés lexicales associées) n'étant pas terminé, ces champs n'apparaissent pas dans la version 3.3, mais seront inclus dans la suivante (Tolone et Voyatzi, 2011).

7.4 Le script de *LGExtract*

Nous commençons par introduire quelques notations complémentaires (7.4.1), avant de montrer quelques extraits des scripts (7.4.2) illustrant le traitement des différents types de propriétés.

7.4.1 Notations complémentaires

Les objets peuvent être paramétrés de deux façons : soit avec la notation **@...@**, comme vu dans la section 7.2, soit avec la notation **@*...@**, permettant de récupérer la valeur dans la table des classes et non dans la classe elle-même. Cette deuxième notation n'est utilisée que pour les constructions, afin de permettre de distinguer les propriétés définitoires codées dans la table des classes des propriétés codées dans les classes. Dans le premier cas, le codage dans la table des classes est une valeur booléenne, alors que dans le deuxième cas, le codage est la valeur o. Cette information est nécessaire lors de la conversion au format Le*fff*, comme nous le verrons en 8.1.1. Par exemple, le code ci-dessous indique, pour la construction N0 V N1 (respectivement, N0 V N1 sur ce point), la valeur booléenne ou o dans la table des classes, suivie de **::**, puis de la propriété N0 V N1 (respectivement, N0 V N1 sur ce point).

```
define abs N0VN1 {construction="@*N0 V N1@::N0 V N1"};
define abs N0VNhumsurcepoint {construction="@*N0 V Nhum sur ce point@::
                                             N0 V Nhum sur ce point"};
```

La propriété N0 V N1 est toujours vraie pour la classe 32H (codage + dans la table des classes) : **@*N0 V N1@** est remplacé par **true**. En revanche, la propriété N0 V Nhum sur ce point dépend des entrées et est codée o dans la table des classes pour la classe 32H et + dans la classe pour l'entrée *alarmer* : **@*N0 V Nhum sur ce point@** est remplacé par **o**. Les deux objets paramétrés montrés ci-dessus, qui seront ajoutés pour cette entrée, deviennent alors :

```
define abs N0VN1 {construction="true::N0 V N1"};
define abs N0VNhumsurcepoint {construction="o::N0 V Nhum sur ce point"};
```

Comme nous l'avons indiqué dans la section 7.2, l'instruction **prop** suivie d'un objet paramétré (notation **@...@**) permet d'indiquer des opérations qui sont appliquées uniquement si la propriété concernée est vraie. Il est possible également de tenir compte d'une propriété fausse (notation **not@...@**), mais avec la contrainte que la colonne doit obligatoirement être codées pour toutes les tables. Par exemple, dans les classes de noms composés, l'entrée est composée de plusieurs mots où il est nécessaire d'indiquer l'ordre dans le script. Cependant, l'ordre dépend de la colonne Adj permut obl (codée dans les tables FNA, FNANA et FNPNA), ce qui oblige l'adjectif à se placer avant le nom si la colonne est codée +. Au total, il y a quatre structures possibles dans toutes les tables de noms :
<ENT>N
<ENT>N <ENT>Adj
<ENT>N <ENT>Prép <ENT>Detc <ENT>Nc

<ENT>Adj <ENT>N

Les trois premières ne posent aucun problème car elles ne se chevauchent pas, en revanche la dernière doit être un cas à part, qui correspond justement au codage + de la colonne Adj permut obl. Le code suivant indique les deux possibilités pour extraire l'entrée complète du nom composé, puis le traitement adéquat selon si la colonne Adj permut obl est codée + ou − :

```
define perm perm [complete="@<ENT>Adj@ @<ENT>N@"];
define notperm notperm [complete="@<ENT>N@ @<ENT>Adj@ @<ENT>Prép@
                                           @<ENT>Detc@ @<ENT>Nc@"];
prop @Adj permut obl@{
  add perm in detN.noun;
}
prop not@Adj permut obl@{
  add notperm in detN.noun;
}
```

7.4.2 Extrait des scripts

Comme nous l'avons expliqué dans la section 7.2, pour chaque catégorie, nous avons un script indiquant les opérations à faire pour chaque propriété. On peut distinguer différents types de propriétés qui sont traités différemment [10] :

– les propriétés concernant l'entrée ou les mots dérivés avec les champs lexicaux associés (par exemple, <ENT>V et <ENT>V-adj), les auxiliaires utilisés pour conjuguer les verbes aux temps composés (Aux =: avoir et Aux =: être), ainsi que les clitiques figés avec le verbe (par exemple, Ppv =: se figé) [11] :

```
prop @<ENT>V@{
  create predV;
  create args;
  add prepositions in predV;
  add locatifs in predV;
  create all-constructions;
  add absolute in all-constructions;
  add relative in all-constructions;
  create example;
}
prop @V-adj@{
  add v-adj in predV;
}
```

10. Les exemples suivants sont tirés des verbes uniquement, les particularités des autres catégories étant détaillées par la suite.

11. Comme déjà vu dans la section 7.2, le code définissant le prédicat verbal **predV**, en l'ajoutant dans **lexical-info** est :
define lexical-info predV [cat="verb",verb=[lemma="@<ENT>V@"], aux-list=()] ;

```
prop @Aux =: avoir@{
  add avoir in predV.aux-list;
}
prop @Ppv =: se figé@{
  add ppvse in predV.verb;
}
```

- les constructions absolues ou complètes qui peuvent être aussi bien la structure de base N0 V N1, que la construction N0 V N1 de N2 codant un argument supplémentaire, ou encore la construction N0 V codant l'effacement d'un argument. Chaque construction est indiquée dans son ensemble, sans faire référence à un argument sous-entendu. On y inclut également les transformations complètes telles que N1 être Vpp par N0, qui, de la même manière, se suffisent à elle-même pour savoir de quels arguments elles sont constituées :

```
prop @N0 V N1@{
  add N0VN1 in absolute;
}
prop @N1 être Vpp par N0@{
  add N1etreVppparN0 in absolute;
}
```

  Notons également que pour certaines constructions, telles que N0 V Prép N1, le programme crée une liste vide de prépositions pour une position donnée (ici, 1, ce qui correspond aux prépositions précédant l'argument N1) [12] :

```
prop @N0 V Prép N1@{
  add preposition1 in prepositions;
  add N0avoirDetNPrepN1 in absolute;
}
```

- les constructions relatives, incluant la pronominalisation (qui remplace un argument par Ppv, comme par exemple dans à N1 = Ppv =: le), les compléments supplémentaires indiqués sans rappeler la construction de base (par exemple, par N4) et les transformations relatives, dont toute la construction n'est pas indiquée. Cela est possible, soit par la présence d'un W désignant d'autres arguments sous-entendus (par exemple, N0 être Vpp W), soit par un code indiquant une transformation spécifique (par exemple, [passif par], qui sous-entend N1 V par N0 ou N1 V par N0 dans N2, etc. en fonction des autres arguments existants puisque tous les arguments sont conservés) :

```
prop @à N1 = Ppv =: le@{
  add aN1Ppvle in relative;
}
prop @par N4@{
  add parN4 in relative;
}
prop @N0 être Vpp W@{
  add N0etreVppW in relative;
}
```

12. Le code associé à **preposition1** est le suivant :
define preposition preposition1 [id="1",list=()] ;

```
prop @[passif par]@{
  add passifpar in relative;
}
```

- la distribution des arguments (par exemple, N0 =: Nhum et N0 =: Qu P) ou leur différentes interprétations, spécifiant le rôle thématique (par exemple, N1 source), où l'on distingue chaque argument (1, 2, 3, etc.) et chaque distribution (trait sémantique humain, complétive, infinitive, etc.) [13] :

```
prop @N0 =: Nhum@{
  add N0 in args;
  add N0hum in N0.dist;
  add orig-N0Nhum in N0hum.origin;
}
prop @N0 =: Qu P@{
  add N0 in args;
  add completive in N0.dist;
  add completiveSubj in N0.dist;
  add orig-N0QuP in completive.origin;
  add orig-N0QuP in completiveSubj.origin;
}
prop @N1 source@{
  add N1 in args;
  add src in N1hum;
  add src in N1-hum;
  add orig-N1source in N1hum.origin;
  add orig-N1source in N1-hum.origin;
}
```

- la distribution des prépositions (par exemple, Prép1 =: à), où l'on distingue chaque argument (1, 2, 3, etc.) et chaque préposition :

```
prop @Prép1 =: à@{
  add a in preposition1.list;
}
```

- la distribution des prépositions locatives (par exemple, Loc N1 =: à N1), où l'on distingue chaque argument (1, 2, 3, etc.) et chaque préposition :

```
prop @Loc N1 =: à N1@{
  add a in locatif1.list;
}
```

Parfois certains types se recoupent. C'est le cas d'une distribution de préposition locative associée à une interprétation de l'argument, spécifiant le rôle thématique (par exemple, Loc N1 =: à N1 destination), ou d'une distribution d'argument associée à une distribution de préposition (par exemple, N1 =: de V0-inf W) :

```
prop @Loc N1 =: à N1 destination@{
  add N1 in args;
```

13. Dans le champs **origin** figurent le ou les intitulés qui sont à l'origine de cette distribution, pour garder un lien direct avec les tables, uniquement à titre informatif.

```
  add destNP in N1.dist;
  add a in destNP.introd-loc;
  add orig-LocN1aN1destination in destNP.origin;
}
prop @N1 =: de V0-inf W@{
  add N1 in args;
  add inf0 in N1.dist;
  add de in inf0.introd-prep;
  add orig-N1deV0-infW in inf0.origin;
}
```

Pour les noms, les mêmes types de propriétés existent avec quelques différences :

– l'entrée pouvant être composée de plusieurs mots, plusieurs propriétés concernes uniquement les mots de l'entrée (<ENT>N, <ENT>Adj, <ENT>Prép, <ENT>Detc, <ENT>Nc), elles sont ajoutées dans **detN.noun** inclus dans **predN** [14] :

```
prop @<ENT>N@{
  create predN;
  add detN in predN;
  add noun1 in detN.noun;
  add Vsup in predN;
...
  add verbales in all-constructions;
  add reductionsGN in all-constructions;
...
}
prop @<ENT>Adj@{
  add adj1 in detN.noun;
}
```

– les constructions sont obligatoirement avec un verbe support (par exemple, N0 avoir Det N Prép N1), qui est ajouté à la liste associée à **Vsup** [15], inclus précédemment dans **predN** (cf. <ENT>N) :

```
prop @N0 avoir Det N Prép N1@{
  add avoir in Vsup.list;
  add preposition1 in prepositions;
  add N0avoirDetNPrepN1 in absolute;
}
```

– les constructions verbales, incluses précédemment dans **all-constructions** (cf. <ENT>N), sont traités séparément (par exemple, N0 V N1) :

```
prop @N0 V N1@{
  add N0VN1 in verbales;
}
```

14. Le code de **predN** et **detN** est :
define lexical-info predN [cat="noun"] ;
define detN detN [noun=[],list-det-modif=()] ;

15. Le code de **Vsup** et de **avoir** est :
define Vsup Vsup [cat="verb",list=()] ;
define Vsuplex avoir value="avoir" ;

– parmi les distributions, nous avons également celles des verbes supports (par exemple, Vsup =: (émettre+formuler)) :
```
prop @Vsup =: (émettre+formuler)@{
  add emettre in Vsup.list;
  add formuler in Vsup.list;
}
```
– parmi les distributions, nous avons également celles du déterminant précédant le nom prédicatif (par exemple, Det =: le), qui est ajouté à la liste associée à **detN**[16], inclus précédemment dans **predN** (cf. <ENT>N) :
```
prop @Det =: le@{
  add le in detN.list-det-modif;
}
```

– nous avons de plus les propriétés de réduction du groupe nominal (par exemple, GN =: le N de N0), incluses précédemment dans **all-constructions** (cf. <ENT>N) :
```
prop @GN =: le N de N0@{
  add leNdeN0 in reductionsGN;
}
```
Pour les expressions figées, il n'y a pas de type différent. On peut cependant remarquer que l'entrée étant composée de plusieurs mots, le nombre de propriétés concernées est encore plus important que pour les noms prédicatifs.

Pour les adverbes, nous avons de plus la structure de l'adverbe (par exemple, Prép1 Det1 C1) :

```
prop @Prép1 Det1 C1@{
  add Prep1Det1C1 in structureAdv;
}
```

7.5 Discussion

Nous effectuons un bilan sur notre outil *LGExtract* (7.5.1), avant de conclure (7.5.2).

7.5.1 Avantages et inconvénients de *LGExtract*

La construction du lexique *LGLex* nous a permis d'identifier clairement les avantages et inconvénients de notre outil. Son avantage principal est l'utilisation de la table des classes. En pratique, toute l'information manquante est rassemblée dans un seul fichier et non dans autant de fichiers que de classes, comme dans l'approche de Gardent *et al.* (2006). De plus, il apporte une vision linguistique plus globale : auparavant, la méthode pour générer un lexique syntaxique à partir des tables du Lexique-Grammaire était de trouver les propriétés définitoires de chaque classe et de les rendre explicites. Maintenant, avec la table des classes, on peut chercher si une propriété donnée est intéressante pour

16. Le code de **detN** (comme vu précédemment) et de **le** est :
define detN detN [noun=[],list-det-modif=()] ;
define det-modif le [det="le+la+l'",modif="false"] ;

une classe donnée, et cela pour l'ensemble des propriétés de toutes les classes de la catégorie concernée. Ceci fait apparaître de nouvelles questions linguistiques dans le cadre des tables du Lexique-Grammaire.

De plus, la combinaison de *LGExtract* avec la table des classes simplifie la maintenance du lexique syntaxique. D'abord, toutes les opérations de chaque propriété sont encodées une seule fois dans le script, indépendamment de chaque classe. Ensuite, s'il survient qu'une nouvelle propriété est constante sur toute une classe, il faut simplement ajouter un symbole + dans la case correspondante de la table des classes. Le script n'a pas besoin d'être modifié pour ajouter cette information dans le lexique généré, étant donné que toutes les opérations concernant cette propriété ont déjà été codées.

Le système nécessite que chaque propriété possède exactement une signification dans toutes les classes. L'emploi de cet outil nous a permis de garder la table des classes cohérente. C'est ainsi qu'auparavant, la propriété zone était une zone de texte dans plusieurs classes mais avec des interprétation différentes :

- dans la plupart des tables, elle fournissait la valeur lexicale des prépositions introduisant les compléments verbaux indépendamment de leur position dans la construction canonique (que nous avons renommée Prép1, Prép2 ou Prép3 selon les cas) ;
- dans la table 38L0, elle indiquait le suffixe à ajouter au verbe afin d'obtenir son nom dérivé (renommée suffixeN) ;
- dans les tables 32R3 et 35R, elle donnait un exemple de complément (renommée thèmeN1).

Nous avons dû ajouter de nouvelles propriétés afin qu'il y en ait une par sens. En particulier, les prépositions ont été numérotées pour qu'il soit possible d'identifier directement les compléments qu'elles introduisent.

Toutefois, certaines limitations sont clairement apparues. Il est parfois nécessaire de répéter des dizaines d'opérations similaires sur des ensembles de propriétés. Par exemple, il est nécessaire de créer manuellement pour chaque propriété des objets linguistiques différents seulement par leur numérotation. Cela est dû au fait que le script ne permet pas de faire des boucles, des fonctions avec paramètres, des tableaux et de la création dynamique d'objets linguistiques. De plus, le programme n'est pas capable d'effectuer des opérations dans un ordre requis. Par exemple, dans les classes dont l'entrée comporte plusieurs mots, comme les classes de noms composés, d'expressions figées ou d'adverbes (semi-)figés, les différents composants de l'entrée sont codés dans plusieurs colonnes successives selon leur ordre d'apparition. Mais il est impossible d'obtenir le nom complet sans en indiquer l'ordre dans le script lui-même (cf. section 7.2). Il serait donc intéressant d'implémenter des macros permettant de telles opérations.

7.5.2 Conclusion

Nous avons développé un outil pour générer des lexiques syntaxiques pour le TAL à partir des tables du Lexique-Grammaire nommé *LGExtract*. Une table des classes est utilisée pour fournir les informations manquantes dans les classes : cela rend explicite toutes les informations implicites sous-jacentes à ces classes. Un script d'extraction par catégorie associe à chaque propriété un ensemble d'opérations qui est activé pour

chaque entrée lorsque la propriété est vraie. Appliqué aux tables du Lexique-Grammaire du français de toutes les catégories (verbes distributionnels, noms prédicatifs, expressions figées et adverbes), cet outil produit un lexique syntaxique exploitable dans les applications de TAL, telles que l'analyse syntaxique.

C'est à partir des entrées de la version textuelle du lexique *LGLex* que nous avons effectué une conversion vers le format Alexina, c'est-à-dire le format du lexique syntaxique Le*fff*, pour permettre son intégration dans l'analyseur syntaxique FRMG.

8

Conversion au format Le*fff*

La création du lexique *LGLex* nous a permis d'envisager l'utilisation des tables du Lexique-Grammaire dans un analyseur syntaxique à grande échelle, l'analyseur FRMG (Thomasset et de La Clergerie, 2005). L'exploitation de l'ensemble des informations lexicales représentées dans les tables du Lexique-Grammaire dans le contexte d'un analyseur syntaxique nécessite de répondre à trois défis :

- représentation entièrement formalisée de ces informations lexicales ;
- couplage avec un analyseur syntaxique ;
- évaluation comparative avec d'autres ressources lexicales, à la fois au niveau des entrées et au niveau des performances de l'analyseur.

Ce chapitre et les suivants décrivent la façon dont nous avons répondu à ces défis, en commençant tout d'abord par la conversion des verbes et des noms prédicatifs du lexique *LGLex* au format Le*fff* (cf. section 4.1), comme nous l'avons décrit dans Sagot et Tolone (2009b); Tolone et Sagot (2011). La section 8.1 présente les fondements linguistiques et la méthodologie pratique de la conversion de *LGLex* au format Le*fff*, en terminant par les problèmes que posent la conversion des expressions figées. Enfin, la section 8.2 présente le lexique obtenu.

8.1 Conversion du lexique *LGLex* en un lexique au format Alexina

Comme nous l'avons mentionné en 2.2.2, la plupart des lexiques syntaxiques ne contiennent que des entrées verbales, à l'exception des tables du Lexique-Grammaire et du Le*fff*, ce dernier étant dans un format TAL directement exploitable dans un analyseur syntaxique, ce qui n'est pas le cas des tables. Le format Le*fff* est donc le plus adapté. C'est pourquoi nous avons réalisé la conversion des tables des verbes simples

et des noms prédicatifs au format Le*fff* à partir du lexique *LGLex*. Le même travail est également en cours pour les tables d'expressions figées (voir 8.1.6) [1]. Le script écrit en Perl effectuant cette conversion se nomme *LGLex2ilex* et fait partie de la distribution actuelle du Le*fff*[2].
La conversion se fait en quatre étapes : identification de la construction de base et de ses variantes (8.1.1), construction des cadres de sous-catégorisation au format Le*fff* (8.1.2), construction des listes de redistributions associées à chaque entrée (8.1.3), ajout aux entrées des informations complémentaires (8.1.4), que nous illustrons par des exemples d'entrées complètes (8.1.5). Enfin, nous détaillons le travail amorcé pour les expressions figées (8.1.6).

8.1.1 Découpage en entrées : la construction de base et ses variantes

Une entrée donnée dans le lexique *LGLex* est associée à diverses constructions. Parmi celles-ci, on peut distinguer plusieurs types de constructions :

1. la construction de base choisie comme référence pour le calcul des autres ;
2. les constructions *de base étendues*, obtenues par adjonction d'arguments à la construction de base. Une grande partie des constructions sont des intermédiaires entre la construction de base et une construction dite *de base maximalement étendue* ou CBME. Par exemple, l'entrée de *rassembler* dans la table 32PL (*Max a rassemblé ses articles (E+dans un ouvrage)*) a pour construction de base la construction transitive simple N0 V N1, mais la possibilité d'ajouter un complément en Loc N2 conduit à une CBME de la forme N0 V N1 Loc N2 ;
3. les constructions qui sont des variantes de la construction de base, obtenues par effacement d'un ou de plusieurs arguments, ou par changement de type de réalisation (Qu P devenant V0-inf W, par exemple) ;
4. les constructions qui sont en réalité des redistributions (constructions relatives telles que [passif de], constructions absolues de type N1 est Vpp de ce Qu P, cf. section 7.3).
5. les constructions qui ne sont pas des variantes de la CBME mais qui ne sont pas considérées comme des redistributions dans le format Le*fff*. Par exemple, dans la table 9, la CBME est N0 V N1 à N2 de Nhum pour certaines entrées. Le complément des constructions N0 V après Nhum et N0 V contre Nhum est incompatible avec les compléments de la CBME :
   Paul écrit à Marie de Luc qu'il est génial

1. Notons que la conversion des adverbes au format Le*fff* n'a pas été envisagée pour le moment, car le Le*fff* contient déjà la liste de tous les adverbes présents dans les tables (sans prendre en compte les variantes, cf. section 7.3 avec pour exemple l'entrée *jusqu'à la fin des temps*) et que FRMG ne fait que les intégrer à tous les endroits possibles dans une phrase sans tenir compte de leur construction syntaxique.

2. Notons qu'il est également disponible sur le site http://infolingu.univ-mlv.fr/ (Données Linguistiques > Lexique-Grammaire > Téléchargement).

Luc écrit contre Bush
**Luc écrit contre Bush à Marie de Luc qu'il est génial*

De par leur origine, les informations présentes dans la section **args** d'une entrée *LGLex* participent à la définition de constructions qui sont des variantes de la construction de base, étendue ou non (cas 2 et 3). Pour identifier les entrées au format Le*fff* à créer à partir d'une entrée *LGLex*, il faut donc identifier, parmi les constructions listées dans la section **all-constructions** la construction de base maximalement étendue (CBME).

Pour identifier la CBME, nous avons développé une méthode pour *aligner* deux constructions, c'est-à-dire construire des correspondances entre arguments, malgré leurs différences de surface (par exemple, Qu P et N1, ou encore à N1 et Prép N1 si l'on sait par ailleurs, grâce à la section **lexical-info** ou à la distribution concernée, que la Prép peut être *à*) et leur possible effacement. Les extensions de la construction de base sont alors identifiées comme étant celles qui s'alignent avec la construction de base [3], à l'exception d'un ou plusieurs arguments supplémentaires. La CBME est alors la construction obtenue en rajoutant à la construction de base l'ensemble des arguments supplémentaires possibles [4].

Nous identifions alors les variantes de cette CBME à l'aide de ce même algorithme d'alignement [5]. Pour chaque variante, nous construisons une séquence d'opérations élémentaires permettant de la reconstituer à partir de la CBME. Ces opérations permettront de construire, pour chaque argument, la liste de ses réalisations possibles, et de déterminer si elles sont effaçables ou non.

À titre d'exemple, voici au format *LGLex* l'entrée du verbe *ruisseler* de la table 35L :

```
ID=V_35L_76;status=to complete
lexical-info=[cat="verb",
              verb=[lemma="ruisseler"],
              aux-list=(avoir="true"),prepositions=(),
              locatifs=(loc=[id="1",list=()],loc=[id="2",list=()])]
args=(const=[pos="0",
             dist=(comp=[cat="NP",nothum="true",
                         introd-prep=(),introd-loc=(),
                         origin=(orig="N0 =: N-hum")])],
      const=[pos="1",
             dist=(comp=[cat="NP",source="true",
                         introd-prep=(),introd-loc=(prep="de",prep="de chez"),
                         origin=(orig="Loc N1 =: de N1 source")])],
```

3. La construction de base fait partie des propriétés précédées de la mention **true::** dans la sous-section **absolute** du lexique *LGLex* (cf. section 7.3). S'il y en a plusieurs, on prend la première apparaissant dans le lexique.

4. La situation est en réalité plus complexe. En effet, certains de ces arguments supplémentaires ne prennent part à aucune des constructions, et ne sont connus que grâce à leur description dans la section **args** de l'entrée. Elles sont prises en compte correctement par notre processus.

5. À l'exception de certaines constructions relatives, telles que Ppv =: y, pour lesquelles la description de la variante correspondante est obtenue directement.

```
    const=[pos="2",
           dist=(comp=[cat="NP",destination="true",
                      introd-prep=(),introd-loc=(prep="vers",prep="dans"),
                      origin=(orig="Loc N2 =: vers N2 destination",
                              orig="Loc N2 =: dans N2 destination")])])
all-constructions=[absolute=(construction="true::N0 V Loc N1",
                             construction="o::N0 V",
                             construction="o::N0 V Loc N1 source Loc N2
                                                               destination",
                             construction="o::N0 être V-ant"),
                   relative=(construction="Ppv =: y",
                             construction="Ppv =: en",
                             construction="[extrap]")]
example=[example="L'eau ruisselle de la gouttière sur les passants"]
```

La construction de base est donc N0 V Loc N1 (cas 1). Les indications sémantiques de type source et destination étant ignorées, la construction N0 V Loc N1 source Loc N2 destination en est l'unique extension, il s'agit donc de la CBME (cas 2). La construction de base devient une variante de la CBME, par effacement de l'argument 2. La construction N0 V est également identifiée comme une variante de la CBME, par double effacement (cas 3). La construction N0 être V-ant est laissée de côté pour l'instant. Du côté des constructions relatives, on trouve la construction Ppv =: y (respectivement, Ppv =: en) qui permettra d'ajouter *y* (respectivement, *en*) à la liste des réalisations de l'argument locatif (respectivement, délocatif) (cas 3). Enfin, la construction [extrap] induira l'adjonction d'une redistribution impersonnelle (*%actif_impersonnel*) à l'entrée (cas 4).

Les constructions qui relèvent du cas 5 reçoivent un traitement par défaut, qui peut être de deux types :

- si l'on veut préserver au mieux le découpage original en entrées, ces constructions donnent lieu à la création de redistributions (au sens du Lefff) *ad hoc* qui ne sont pas définies formellement ;
- si l'on veut préserver l'utilisabilité dans les outils automatiques (compilation à la Lefff puis utilisation dans un analyseur syntaxique), ces constructions donnent lieu à la création d'entrées distinctes, dites *entrées secondaires* (voir l'option *-nuc* expliquée dans l'annexe D).

Notons que les entrées secondaires crées ont le même identifiant. Le découpage original peut donc être retrouvé, en associant à chaque identifiant (et non entrée) l'ensemble des constructions.

Pour certaines constructions, il semble qu'elles auraient dû conduire à créer des entrées secondaires avec un nouvel identifiant car leur sens est différent. Par exemple, dans la table 13, la construction de base est N0 V N1 de N2 et certaines entrées acceptent la transformation N1 se V de ce Qu P :
Max félicite Luc qu'il ait réussi à séduire Léa
Luc se félicite d'avoir réussi à séduire Léa (se féliciter = se réjouir)

Mais par exemple, pour l'entrée *instruire*, le sens reste le même :
Jo a instruit Luc de ce qui s'est passé
Luc s'est instruit de ce qui s'est passé
Cela montre qu'il n'y a pas besoin de créer d'entrée secondaire avec un identifiant différent, mais plutôt qu'il faudrait retirer des tables les codages + qui ne respectent pas le principe d'avoir «un sens par entrée».

8.1.2 Construction des cadres de sous-catégorisation

Une fois répertoriées les entrées à produire, les cadres de sous-catégorisation sont construits. Pour cela, on construit d'abord le cadre correspondant à la construction maximale de chaque entrée (la CBME pour l'entrée *canonique*, ou l'unique construction des entrées secondaires). Pour cela, on utilise des heuristiques permettant de définir la fonction syntaxique de chaque argument, ainsi que sa réalisation canonique.

Les fonctions syntaxiques sont obtenues de la façon suivante. Tout d'abord, le premier argument reçoit toujours la fonction *Suj*[6]. Le premier argument post-verbal, s'il est direct, se voit attribuer la fonction *Obj*, sauf pour les entrées de la table 32NM[7]. Ensuite, un argument introduit par *à* (respectivement, *de*) reçoit la fonction syntaxique *Objà* (respectivement, *Objde*), sauf si un indice complémentaire vient contredire ce choix[8]. Les arguments introduits par Loc ont la fonction syntaxique *Loc*, sauf ceux de la forme Loc Ni source ou vérifiant Loc Ni =: de Ni source, qui ont la fonction syntaxique *Dloc*. Enfin, les autres arguments sont considérés comme des *Att* s'ils sont directs, et comme des *Obl* s'ils sont introduits par une préposition (*Obl2* si un *Obl* existe déjà).

Pour les noms prédicatifs, le même ensemble de fonctions est utilisé, mais le verbe est remplacé par un verbe support suivi d'un groupe nominal n'ayant pas de fonction syntaxique.

Les réalisations de ces fonctions sont construites en deux temps. Tout d'abord, le type de syntagme (nominal, infinitif, phrastique, etc.) est déterminé. La réalisation *scompl* correspond aux distributions de catégorie **comp** et **ceComp** ainsi qu'aux arguments dans les constructions en Qu P. La réalisation *sinf* correspond aux distributions de catégorie **inf** et aux arguments en Vi-inf W. La réalisation *qcompl* correspond aux distributions de catégorie **siPOuSiP**. La réalisation *sn* correspond aux distributions de catégorie **NP** et **leFaitComp** et aux arguments en **Ni**. Enfin, la réalisation *sa* correspond aux distributions de catégorie **adj**. De plus, la réalisation *cln* (clitique nominatif) est ajoutée systématiquement à la fonction syntaxique *Suj*.

Ceci accompli, il reste à lister les introducteurs possibles. L'ensemble des prépositions et autres introducteurs (par exemple, *et*) sont pris en compte. En particulier, la

6. Les constructions impersonnelles sont toutes obtenues sous forme de redistributions, puisque la table 31I ne fait plus partie des entrées verbales du lexique *LGLex*.

7. Cette table rassemble les verbes à pseudo-objet de type *peser* (*Le sac pèse 10 kg*) ou *sentir* (*Cette pièce sent la fumée*).

8. Par exemple, pour un argument N1 introduit par *à*, la propriété à N1 = Ppv =: le lui conférera la fonction syntaxique *Obj* (comme dans *Il apprend à conduire / Il l'apprend*), alors que la propriété à N1 =: y lui conférera la fonction syntaxique *Loc*.

préposition est parfois Prép. Il faut alors extraire la liste des prépositions possibles de la section **lexical-info** de l'entrée *LGLex* (dans le champ **prepositions**) et/ou dans la distribution concernée (dans le champ **introd-prep**). Lorsque la préposition est Loc, elle peut être précisée par les mêmes moyens que Prép (quoique dans des champs distincts : **locatifs** et **introd-loc**), mais ce n'est pas obligatoire (toute préposition locative étant alors admise).

Le résultat de ces heuristiques est le cadre de sous-catégorisation au format Le*fff* pour la construction maximale de l'entrée. Ainsi, la CBME de l'exemple précédent devient :

```
<Suj:sn|cln,Dloc:sn,Loc:sn>
```

Les constructions maximales des entrées qui sont canoniques (leur construction maximale est une CBME) doivent alors être complétées. Pour cela, nous ajoutons tout d'abord les réalisations issues de constructions telles que Loc N1 = Ppv =: y. On peut avoir *cln* pour le clitique nominatif, *cla* pour le clitique accusatif, *cld* pour le clitique datif, *y* pour le clitique locatif, *en* pour le clitique génitif.

Nous répercutons ensuite pour chaque variante de la CBME la séquence d'opérations élémentaires permettant de les dériver de la CBME de la façon suivante : tout effacement d'un argument rend l'argument facultatif ; tout changement de réalisation induit une nouvelle réalisation possible de la fonction syntaxique concernée.

Les dépendances entre ces opérations (par exemple, tel argument est effaçable seulement si tel autre l'est aussi) sont *perdues* : tout effacement est considéré comme possible sans condition, alors même qu'il peut ne venir que d'une seule construction. De même, toute réalisation d'un argument autorisée par une construction devient possible quelles que soient les réalisations des autres arguments. Il s'agit là formellement d'une approximation des données linguistiques présentes dans les tables (et dans le lexique *LGLex*). Cette approximation a le mérite de permettre de diminuer au maximum le nombre d'entrées, même s'il reste à démontrer qu'elle n'a pas de conséquences pour l'utilisation du lexique produit, puisqu'en théorie les dépendances entre effacements permettent d'éliminer des analyses et donc de lever des ambiguïtés. Cette approximation est également adoptée par le modèle de la valence mis en œuvre par le lexique Dicovalence (van den Eynde et Mertens, 2006).

Le résultat de ce processus est un cadre de sous-catégorisation, avec alternatives et arguments optionnels. Le cadre construit pour l'entrée canonique de l'exemple précédent est alors :

```
<Suj:cln|sn,Dloc:(de-sn|de_chez-sn|en),Loc:(vers-sn|dans-sn|y)>
```

8.1.3 Identification des redistributions admissibles

Le cadre de sous-catégorisation de base construit à la section précédente est le cadre dit *profond*, car il relève de la syntaxe profonde. Néanmoins, pour les verbes, la table 31I des verbes impersonnels intrinsèques et autonomes faisant partie des expressions figées, ce cadre profond correspond toujours à un cadre de surface, celui de la (re)distribution

active (*%actif*). Toutes les entrées verbales se voient donc attribuer au moins cette (re)distribution.

Les autres redistributions admissibles sont identifiées parmi les constructions répertoriées dans la section **all-constructions** de l'entrée *LGLex* :

- [passif par] et [passif de] correspondent aux redistributions %passif (*Jean assistait Max depuis des années → Max était assisté par Jean depuis des années*) et %passif_de (*Marie aime Pierre → Pierre est aimé de Marie*) ;
- [extrap] correspond à la redistribution %actif_impersonnel, c'est-à-dire, constructions impersonnelles à l'actif avec le sujet inversé, le cas échéant (*Un accident est arrivé à Jean → Il est arrivé un accident à Jean*) ;
- il être Vpp Prép N1 (E + par N0) (anciennement [extrap][passif]) correspond à la redistribution %passif_impersonnel, c'est-à-dire, constructions impersonnelles au passif avec le sujet inversé, le cas échéant (*Cette nouvelle information clarifie pourquoi Max est fâché → Il est clarifié par cette nouvelle information pourquoi Max est fâche*).

De plus, on a la redistribution %ppp_employé_comme_adj pour les participes passés employés comme adjectifs (*Marie s'assoie → Marie est assise*), qui est appliquée par défaut à tous les *Obj* puisque l'information n'est pas codée dans les tables.

Comme cela a été étudié par Danlos et Sagot (2008), les redistributions pronominales (de type *se* moyennes et *se* neutres) étant mal codées dans le Lexique-Grammaire, nous ne les avons pas prises en compte en tant que redistributions.

Quant aux entrées nominales, on leur attribut uniquement la redistribution %default, qui construit un cadre de sous-catégoriation final identique à l'initial. De plus, elles peuvent avoir la redistribution %passif (*Max fait une déposition contre Luc auprès du commissariat → Une déposition a été faite contre Luc auprès du commissariat*).

Le résultat de cette extraction pour l'exemple précédent est ainsi :

```
<Suj:cln|sn,Dloc:(de-sn|de_chez-sn|en),Loc:(vers-sn|dans-sn|y)>;;%actif,
%actif_impersonnel
```

Un exemple plus riche, tel que celui de l'entrée de *clouer* dans la table 36SL (*Max a cloué cette planche au mur*), devient quant à lui :

```
<Suj:cln|sn,Obj:sn,Loc:(avec-sn|et-sn|à-sn|sur-sn)>;;%actif,
%passif,%ppp_employé_comme_adj
```

8.1.4 Prise en compte des informations complémentaires

D'autres types d'informations sont alors ajoutés, pour former l'entrée finale. Tout d'abord, le prédicat sémantique correspondant à l'entrée, qui est le plus souvent *Lemma* (c'est-à-dire identique au lemme verbal), peut être complété en fonction d'informations présentes dans la section **lexical-info** de l'entrée *LGLex* (par exemple, **ppvse="true"**, **neg="true"**, etc. donne *se Lemma*, *ne pas Lemma*, etc.). Le lemme lui-même, en-tête de l'entrée, est extrait du champ **lemma**, et complété par le numéro de la table dont est

issue l'entrée, ainsi que du numéro d'entrée dans cette table. Ainsi, même lorsqu'une entrée *LGLex* donne lieu à plusieurs entrées au format Le*fff*, ces entrées partagent un même identifiant.

Les informations syntaxiques complémentaires les plus intéressantes sont représentées sous forme de macros. Il s'agit des informations suivantes :
- auxiliaire de conjugaison du verbe, récupéré dans le champ **aux-list**, qui induit l'ajout de la macro @*avoir* (par exemple, pour le verbe *achever* dans *Max a achevé de peindre le mur*) ou de la macro @*être* (par exemple, pour le verbe *s'arrêter* dans *Max s'est arrêté de boire*) ;
- caractère (essentiellement) pronominal du verbe : **ppvse="true"**, macro @*pron* (par exemple, pour le verbe *arrêter* dans l'exemple précédent) ;
- caractère obligatoirement négatif du verbe : **neg="true"**, macro @*neg* (par exemple, pour le verbe *rajeunir* dans *Que sa fille ait 20 ans ne rajeunit pas Max*) ;
- autres clitiques obligatoirement figés au verbe : **ppven="true"**, macro @*pseudo-en* (par exemple, le verbe *baver* dans *Max en bave avec Luc*), **ppvy="true"** ; macro @*pseudo-y* (par exemple, le verbe *aller* dans *Luc y va*) ; **ppvle="true"**, macro @*pseudo-le* (par exemple, le verbe *disputer* dans *La haine le dispute à la colère*) ; **ppvla="true"**, macro @*pseudo-la* (par exemple, le verbe *fermer* dans *Fermez-la*) ; **ppvles="true"**, macro @*pseudo-les* (par exemple, le verbe *aligner* dans *Max les aligne à Luc*) ;
- mode des complétives possibles (champ **mood** de la distribution correspondante), indiquée sous la forme de macros de type @fM, où f vaut respectivement *SComp*, *Comp*, *AComp* et *DeComp* pour les complétives de fonction syntaxique *Suj*, *Obj*, *Objà* et *Objde*, et où M est *Ind* ou *Subj*. Par exemple, @*SCompInd* indique que si le sujet est réalisé par un syntagme phrastique fini, son mode est l'indicatif (par exemple, *qu'il fait beau* dans *Pierre dit qu'il fait beau*) ; au contraire, @*CompSubj* indique que si l'objet direct est réalisé par un syntagme phrastique fini, son mode est le subjonctif (par exemple, *qu'il fasse beau* dans *Pierre veut qu'il fasse beau*) ;
- informations de contrôle, extraites du champ **contr** des distributions ou du nombre i dans les arguments de type Vi-inf au sein des constructions, et qui sont représentées par des macros de la forme @*Ctrl*$f_1 f_2$, où f_1 est la fonction syntaxique du contrôleur et f_2 celle du contrôlé. Par exemple, @*CtrlSujObj* indique que s'il est réalisé par un syntagme infinitif, l'objet est contrôlé par le sujet (par exemple, *chercher du pain* est contrôlé par *Pierre* dans *Pierre va chercher du pain*).
- trait sémantique humain ou non humain pour le syntagme nominal (champ **cat="NP"** avec le champ **hum="true"** ou **nothum="true"** de la distribution correspondante), indiqué sous la forme de macros de type @fM, où f est la fonction syntaxique et M est *Nhum* ou *N-hum*. Par exemple, @*ObjàNhum* indique que l'objet indirect introduit par la préposition *à* peut dénoter une personne ou un animal linguistiquement assimilé à une personne lorsqu'il est réalisé par un syntagme nominal (par exemple, *Vercingetorix s'est rendu à l'ennemi*) ; au contraire, @*ObjàN-hum* indique que l'objet indirect introduit par la préposition *à* peut dénoter un non humain (par exemple, *Jean s'est rendu à mon opinion*) ; dans

ce cas, *@ObjàNhum* et *@ObjàN-hum* ne font pas partie de la même entrée car le sens de *se rendre* est différent ; dans d'autres cas, par exemple, *Max va tomber* et *Le verre va tomber*, *@SujNhum* et *@SujN-hum* indiquent que le sujet peut dénoter un humain et un non humain pour la même entrée ; si aucune indication n'est donné pour un argument, on peut considérer que les deux sont toujours possibles.

Ce dernier type de macros, n'existant pas dans le Le*fff*, il a été créé afin de représenter ce trait sémantique présent dans l'ensemble des tables. Les autres traits sémantiques, n'étant pas codés systématiquement, n'ont pas été gardés, mais l'on peut envisager d'ajouter d'autres macros si nécessaire.

Enfin, la table de conjugaison devant être associée à l'entrée est récupérée dans le Le*fff*, permettant ainsi la compilation du lexique obtenu avec la description morphologique du français du Le*fff*.

Pour les verbes, les différentes classes sont *v-er:std* pour les verbes du premier groupe, y compris certains verbes qui se terminent en *eler* et *-eter* (*acheter : j'achète*), *v-er:dbl* pour les verbes qui se terminent en *eler* et *-eter* qui doublent le *l* ou le *t* (*jeter : je jette*), *v-ir2* pour les verbes du deuxième groupe (*finir*), *v-ir3* pour les verbes réguliers du troisième groupe qui se terminent en *-ir* (*servir*), *v-re3* pour les verbes réguliers du troisième groupe qui se terminent en *-re* (*vendre, mordre, vaincre, ébattre, rompre*), puis des classes spécifiques pour les verbes très irréguliers (par exemple, *v67* pour les verbes qui se terminent en *-paraître*, *v79* pour ceux qui se terminent en *-ure*, etc.), ainsi que des classes spécifiques pour chaque verbe ayant une conjugaison unique tels que *v-aller* pour *aller*, *v-faillir* pour *faillir*, etc.

Dans les cas où un lemme verbal n'est pas présent dans le Le*fff*, il lui est attribué la classe standard des verbes du premier groupe (*v-er:std*) s'il se termine en *-er* ; l'unique classe des verbes du deuxième groupe (*v-ir2*) s'il se termine en *-ir* ; une des classes principales des verbes du troisième groupe (*v-re3*) s'il se termine en *-vendre*[9] ; et la classe correspondant au verbe *paraître* (*v67*) s'il se termine en *-paraître*. Dans les autres cas, un avertissement est émis et l'entrée est ignorée. Dans la pratique, les très rares cas initialement concernés ont fait l'objet pour la plupart d'un ajout manuel dans le Le*fff*, ainsi presque plus aucun verbe ne rentre dans cette catégorie[10]. De plus, ceci a permis la détection d'erreurs dans les entrées des tables, qui ont été également corrigées[11]. Par exemple, *<E>liger* pour *négliger* dans la table 6, et *en finir* pour *finir* avec l'ajout de *en* dans la colonne Ppv de la table 1. C'est également le cas de noms dans les tables verbales, comme *charge* pour *charger* dans la table 32R3, *fossoyeur* pour *fossoyer* dans

9. Il faudrait également associer à cette même classe les verbes se terminant par *-tendre*, afin de prendre en compte l'entrée *sous-tendre*.

10. Les 2 seules entrées concernées sont *sous-tendre*, comme nous venons de le mentionner, et *contrire*, dont la conjugaison est difficile à trouver puisque ce verbe n'est présent que dans le Dictionnaire du Moyen Français.

11. Une comparaison des entrées verbales entre la première version de 2008 et la version 3.3 est réalisée à la section 9.2. Cela ne tient pas compte d'erreurs pouvant avoir été introduites par nous-même lors de la modifications des tables et ayant été corrigées par la suite.

la table 38LD, et *rencontre* pour *rencontrer* dans la table 38LR.

Pour les noms, les classes principales sont celles qui produisent deux formes, une au singulier et une au pluriel en ajoutant simplement un *s*, que ce soit pour les noms masculins (*nc-2m*) ou féminin (*nc-2f*). La majorité des noms se terminant en *-u*, prennent un *x* au pluriel (*nc-u2m*). Quelques noms n'ont qu'une seule forme car ils s'écrivent de la même manière au singulier qu'au pluriel (*nc-1m* ou *nc-1f*). D'autres encore n'ont qu'une forme au pluriel (*nc-1mp* ou *nc-1fp*). Ensuite, certains noms peuvent s'employer aussi bien au féminin qu'au masculin, avec pour chacun les deux formes singulier et pluriel, ils ont donc quatre formes au total (*nc-4*). D'autres, se terminent en *-s* et n'ont que trois formes car la forme au masculin singulier et masculin pluriel est identique (*nc-s3*). Les noms non trouvés dans le Le*fff*, sont considérés comme invariables (*inv*), le nom n'est donc pas fléchi.

8.1.5 Exemples d'entrées complètes

Pour l'entrée *ruisseler* de la table 35L, que nous suivons depuis le début de ce chapitre, le résultat final est alors le suivant :

```
ruisseler___V_35L_76 v-er:dbl
                     100;Lemma;v;
                     <Suj:cln|sn,Dloc:(de-sn|de_chez|en),Loc:(vers-sn|dans-sn|y)>;
                     cat=v,@SujN-hum;
                     %actif,%actif_impersonnel
```

Pour l'entrée *clouer* de la table 36SL, mentionnée ci-dessus, le résultat est :

```
clouer___V_36SL_28 v-er:std
                   100;Lemma;v;
                   <Suj:cln|sn,Obj:sn,Loc:(avec-sn|et-sn|à-sn|sur-sn)>;
                   cat=v,@ObjN-hum,@SujN-hum,@SujNhum;
                   %actif,%passif,%ppp_employé_comme_adj
```

Cette entrée permet de reconnaître par exemple *Max a cloué cette planche au mur*, mais aussi *Max a cloué cette planche*.

Prenons comme exemple, le nom prédicatif *considération* de la table AA. Il a deux constructions définitoires comme on peut le voir dans cet extrait du lexique *LGLex* :

```
absolute=(construction="true::N1 avoir Det N de N0",
          construction="true::N0 avoir Det N Prép N1")
```

Lors de la conversion au format Le*fff*, cela crée deux entrées avec le même identifiant. La catégorie morphologique est nom commun (*cat=nc*) et la catégorie syntaxique est *cf*, désignant un constituant figé séparable du verbe. Dans *lightverb*, on retrouve la liste des verbes supports, et la redistribution est *%default* :

```
considération___N_aa_18 nc-2f
                        100;Lemma;cf;
                        <Suj:cln|sn,Obl:pour-sn>;
                        cat=nc;@SujNhum;@OblN-hum;@OblNhum;
                        lightverb=avoir|ressentir|éprouver;
                        %default
considération___N_aa_18 nc-2f
                        100;Lemma;cf;
                        <Suj:cln|sn,Objde:de-sn>;
                        cat=nc;@SujNhum;
                        lightverb=avoir;
                        %default
```

La première entrée permet de reconnaître *Max a de la considération pour (ce geste+Luc)* et la deuxième *(Ce geste+Luc) a la considération de Max*.

Pour récapituler, une entrée du lexique **intensionnel**[12] contient les informations suivantes :
- l'identifiant de l'entrée : catégorie_numTable_numEntrée ;
- une classe morphologique, qui définit le modèle qui construit les formes fléchies en reposant sur les classes flexionnelles du Le*fff* ;
- un poids, qui désigne la fréquence d'apparition d'un mot dans la langue française (dans le but d'aider l'analyseur syntaxique à privilégier une entrée plutôt qu'une autre) ;
- la forme du lemme incluant les formes préverbales ;
- une catégorie syntaxique (*v*, *cf* pour un constituant figé séparable du verbe, ou *cfi* pour un constituant figé inséparable du verbe) ;
- une catégorie morphologique (*cat=v* ou *cat=nc*) ;
- un cadre de sous-catégorisation initial ;
- des informations supplémentaires représentées par des macros ;
- une liste des redistributions possibles.

Pour terminer, nous donnons un exemple d'une entrée du lexique **extensionnel** : le participe passé féminin pluriel (*Kfp*) du verbe *clouer* associé à la redistribution passive permettant de reconnaître *Cette planche a été clouées au mur par Max* :

```
clouées 100 v [pred="clouer___V_36SL_28__1
              <Suj:sn,Loc:(avec-sn|et-sn|sur-sn|à-sn),Obl2:(par-sn)>",
              @passive,@pers,cat=v,@Kfp]
              clouer___V_36SL_28__1 PastParticiple Kfp %passif
```

L'objet direct original (*Obj*) a été transformé par un sujet passif et un agent optionnel (*Obl2*), réalisé par un syntagme nominal précédé par une préposition (*par-sn*), a été ajouté.

12. Afin de voir d'autres exemples, la section 9.3 montre les entrées verbales *croupir*, *croustiller* et *croûter* de la table 31R dans tous les formats disponibles.

Pour plus de détails, voir l'annexe C.

8.1.6 Conversion des expressions figées

Nous décrivons d'abord la méthode générale adoptée, avant d'évoquer les cas particuliers de certaines tables.

Méthode générale

La conversion des expressions figées a d'abord été faite en ne prenant en compte que la construction de base, ce qui a le mérite de couvrir la grande majorité des intitulés des tables, lesquels représentent les différents mots de l'entrée, ainsi que les distributions des constituants libres. Les variantes de constructions et les redistributions ont donc été ignorées.

La conversion des expressions figées du Lexique-Grammaire a déjà été étudié par Laurence *et al.* (2006), ce qui a permis l'intégration de 1 915 expressions figées dans le Le*fff* (*vC1pC2.ilex*, *vC1pN2.ilex*, *avoirC1pC2.ilex*, *avoirN1pC2.ilex*). Voici un exemple d'expression figée du Le*fff*, sur lequel nous nous sommes basée (*vC1pC2.ilex*) :

```
allumer une flamme de colère dans le regard 0+
                                           100;Lemma;cf;
                                           <Suj:cln|sn,Objde:de-sn|en>;
                                           synt_head=$1;
                                           %default
```

Il importe de noter qu'il y a également dans le Le*fff* des exemples avec verbe support qui pourrait être utilisé comme modèle pour certaines tables en *avoir* ou *être*, mais ce n'est pas le cas puisqu'on ne les considère pas comme des noms prédicatifs où plusieurs verbes supports seraient possibles, mais comme une expression figée entière. À la différence des tables de noms prédicatifs, le verbe est ici également dans l'entrée (*avoirC1pC2.ilex*) :

```
avoir affaire à la justice 0+
                           100;Lemma;cf;
                           <Suj:cln|sn>;
                           lightverb=avoir;
                           %default

avoir de l'argent plein les poches 0+
                                   100;Lemma;cf;
                                   <Suj:cln|sn>;
                                   lightverb=avoir;
                                   %default
```

Les valeurs par défaut pour les expressions figées sont :

- code de flexion (ou classe morphologique) : *0+* ; le *0* permet de supprimer le premier mot (car il est en première position, *inv+0+* permet de supprimer le deuxième mot, etc.), c'est-à-dire le verbe. En effet, il peut s'insérer des adverbes par exemple entre le verbe et le reste, donc le verbe est dans l'entrée lexicale intensionnelle mais pas extensionnelle. Une séquence se finissant par un + signifie que tous les autres mots sont invariables, sans avoir à écrire *0+inv+inv+inv* avec le bon nombre de mots. S'il y a des clitiques figés avec le verbe, on ajoute autant de *0* qu'il y a de clitiques (par exemple, pour l'entrée *n'en être pas moins vrai*, le code de flexion est *0+0+0+* ;
- pas de catégorie morphologique ;
- catégorie syntaxique : *cfi* (constituant figé inséparable du verbe) ;
- tête syntaxique : *synt_head=$1*, ce qui correspond au premier mot, c'est-à-dire le verbe. Au niveau extensionnel, c'est le trait *synt_head* qui indique quels sont le ou les verbes possibles comme tête du reste pour former la locution figée complète. Un problème se pose lorsque le premier mot est la négation ou Ppv : il faut rajouter des accolades autour du verbe pour qu'il soit considéré comme premier mot ;
- redistribution par défaut : *%default*.

A propos des entrées, voici ce qui a du être rajouté :
- suppression des <*E*> présents dans l'entrée. Au maximum 12 suppressions sont possibles puisque la plus longue entrée des expressions figées comporte 12 mots ;
- suppression des espaces au début de l'entrée. En effet, dans les tables, beaucoup de mots sont introduits par un espace inutile, mais ceci serait trop long à supprimer manuellement ;
- suppression des espaces suivant une apostrophe : c'est le cas pour les négations en *n'* ;
- élision de *le*, *la* et *de*, devant une voyelle ou un *h*. En effet, les déterminants ne tiennent pas compte du mot qui suit. Cette approximation est fausse pour la soixantaine de mots commençant par un *h* aspiré, malheureusement il n'existe pas d'autre moyen pour les reconnaître.

Cela est sans compter que les entrées n'ont pas encore été corrigées et que certaines contiennent des codes de remplacement, comme par exemple *:Nhum* ou le déterminant *Poss0*. Pour ce dernier, on peut ajouter une macro de type *@ObjPossSuj* qui indiquerait que si le déterminant du *Obj* est un *Poss*, il fait référence à l'argument de fonction syntaxique *Suj*, c'est-à-dire de position *0*. Ces entrées posent problème pour la conversion au format Le*fff* tant qu'un traitement adapté à chaque code de remplacement n'a pas été établi. De plus, les notations de ces codes ont besoin d'être harmonisées (par exemple, *lui-0* et *Lui-0*).

Pour reconnaître ce qui est figé dans la construction de base, il a fallu lister explicitement les motifs (par exemple Det1 C1) à ne pas prendre en compte dans le traitement, puisqu'ils ne donnent aucune autre information que l'ordre des mots, ce qui est déjà contenu dans l'entrée. Il ne faut garder que les informations sur le verbe et sur les arguments libres tels que Prép2 N2 ou Prép3 N3. Les mots de l'entrée sont codés dans **lexical-info**, y compris les prépositions des arguments libres. La préposition du dernier

argument libre est donc récupérée et ensuite supprimée de l'entrée. Elle constitue en effet le dernier mot de l'entrée, alors que la préposition est codée dans le cadre de sous-catégorisation. La distribution de l'argument N2 ou N3 en question est récupérée par le procédé habituel dans **args**.

Pour les prépositions libres, il y avait confusion entre les prépositions vides et la non existance de l'argument concerné. C'est pourquoi la construction de base de la table EC0 qui était Ce être Prép1 Det1 C1 Prép2 Det2 C2 Prép3 N3 alors que le troisième argument n'était pas obligatoire (*c'est une affaire qui marche*), a été changée en Ce être Prép1 Det1 C1 Prép2 Det2 N2 avec l'ajout de la colonne Ce être Prép1 Det1 C1 Prép2 Det2 C2 Prép3 N3 dans la table qui vaut + lorsqu'au moins l'une des 5 colonnes N3 =: Nhum, N3 =: N-hum, N3 =: ce Qu Pind, N3 =: ce Qu Psubj et N3 =: (de) V-inf W vaut +. De la même manière, les constructions définitoires de la table Z5P étaient N0 être Prép1 Det1 C1 (E+Prép2 N2) et il être Prép1 Det1 C1 (E+Prép2 N2) Qu P alors que le deuxième argument n'était pas obligatoire, elles ont donc été changées en N0 être Prép1 Det1 C1 et il être Prép1 Det1 C1 Qu P. Cela étant accompagné de l'ajout des 2 colonnes N0 être Prép1 Det1 C1 Prép2 N2 et il être Prép1 Det1 C1 Prép2 N2 Qu P qui valent toutes les 2 + lorsque l'une des 2 colonnes N2 =: Nhum et N2 =: N-hum vaut +.

Après ces quelques modifications, on peut établir la règle que lorsqu'une préposition vaut <*E*>, cela signifie qu'il n'y a pas de préposition introduisant l'argument, mais que l'argument existe quand même, nous l'avons noté <*Obl:sn*>. Ceci concerne les tables A1PN, CPQ, EPCPN et EPCPQ avec des entrées comme *en être à croire Nhum*.

Actuellement, 60 tables, sur 69 au total, traités de cette manière donne les bonnes entrées correspondant à la construction de base de la table. Pour beaucoup de constructions, le cadre de sous-catégorisation n'est constitué que du sujet (<*Suj:cln|sn*>) car tout le reste est figé, seul le sujet est libre.

Cas particuliers

Justifions à présent les spécificités des entrées de certaines tables :

- Pour les tables C5, CAN, CDN, CPN, EDN et EPCDN, les compléments de noms (Nc) sont convertis en deuxième argument (N2), car il n'est pas possible dans le Le*fff* de coder les compléments de noms. Par exemple, la construction N0 V Prép1 Det1 C1 de Nc code un *Objde :de-sn*, c'est-à-dire un syntagme nominal libre ;
- De plus, pour la table CAN qui possède 2 constructions définitoire, une seule construction n'est prise en compte comme construction de base, à savoir N0 V Det1 C1 de Nc, la deuxième (N0 V Det1 C1 à Nc) doit par la suite être prise en compte parmi les variantes possibles. De même, pour la table Z5P, comme cela a été décrit précédemment, c'est uniquement il être Prép1 Det1 C1 Qu P qui est prise en compte, alors qu'il faudrait pouvoir prendre en compte N0 être Prép1 Det1 C1 ;
- Pour la table EC0, comme vu précédemment, l'unique construction de base prise

en compte est Ce être Prép1 Det1 C1 Prép2 Det2 C2, ce qui signifie que contrairement à ce qui est expliqué, la préposition 3 ne peut pas être supprimée de l'entrée (par exemple, *c'est tout pour*) puisqu'il manque l'information indiquant qu'elle fait partie d'un troisième argument qui est libre ;

- Pour les tables ANP2, C6, CNP2 et ENPC, la construction de base contient un argument libre (N1) directement après le verbe, comme dans N0 V N1 Prép2 Det2 C2. La catégorie syntaxique est donc *cf* (constituant figé séparable du verbe) au lieu de *cfi*, ce qui permet d'insérer des mots après le verbe. Cela est possible grâce au code de flexion *0+* qui spécifie que le verbe est supprimé, fléchi et peut être suivi par d'autres mots. La distribution du N1 contenue dans **args** doit être ignorée puisque cet argument ne peut pas faire partie du cadre de sous-catégorisation s'il n'est pas placé après l'entrée complète ;
- De plus, pour la table ENPC, ayant pour construction de base N0 être Det1 N1 Prépc Detc Cc, l'argument Det1 N1 est supprimé, avec y compris l'information sur le Det1 qui est figé. Notons également que la construction de base N0 être Det1 N1 Prépc Detc Cc est incompatible avec les 2 colonnes codées dans la table (N0 être Det1 Prépc Detc Cc N1 et N0 être Det1 N1 Prépc Detc Cc) qui codent l'ordre des mots, qui peut être différent de celui de la construction de base. La deuxième colonne dans la table n'est donc pas prise en compte puisqu'il s'agit de la construction de base. Étant souvent codée + et pouvant être poussée lorsque l'autre ordre est accepté, cela reste cohérent ;
- Pour les tables C0, C0E, C0Q et certaines entrées de la table CFF, le sujet est figé : pour les tables C0 et C0Q, le sujet est Det0 C0 (dans la construction de base Det0 C0 V Prép1 (Det1 C1+N1) Prép2 N2), pour la table C0E, Conj Prép0 Det0 C0 (dans construction de base Conj Prép0 Det0 C0 Prép1 (Det1 C1+N1) Prép2 N2), et pour la table CFF, Conj Det0 C0 pour certaines entrées (dans construction de base Conj (Det0 C0+N0) V (Det1 C1+N1) Conj-2 Det0-2 C0-2 V-2 Det1-2 C1-2). Les entrées sont donc des entrées verbales codées comme par exemple *voici* (*interj.ilex*) :
```
voici inv
      100;Lemma;v;
      <Obj:cla|qcompl|scompl|sinf|sn>;
      cat=v;%actif
```
  Toutes les valeurs par défaut sont donc modifiées : la catégorie syntaxique est *v*, la catégorie morphologique est *cat=v*, la redistribution par défaut est *%actif*, et le cadre de sous-catégorisation ne contient pas de sujet, mais peut contenir un ou deux compléments lorsque l'argument N1 et/ou Prép2 N2 est libre.
  Pour les entrées avec sujets figés, le code de flexion se compose de plusieurs *inv* (pour chaque mot du sujet figé, deux pour les tables C0 et C0Q correspondant à Det0 C0, et trois pour les entrées de la table CFF acceptant Conj Det0 C0), puis d'autant de *0* que de clitiques figés avec le verbe, suivi du code de flexion du verbe. La table C0E n'ayant pas de verbe, son code de flexion est *inv+* ;
- Pour les tables E01, E0P1 et EC0, le sujet est également figé, mais en fait c'est l'entrée toute entière qui est figée : pour les tables E01 et E0P1, le sujet est Det0 C0 (dans les constructions de base Det0 C0 être Adj1 et

Det0 C0 être Prép1 Det1 C1), et pour la table EC0, Ce (dans la construction de base Ce être Prép1 Det1 C1 Prép2 Det2 C2). Les entrées sont donc des entrées verbales codées comme par exemple *avoir___AUX* (*v_new.ilex*) :

```
avoir___AUX v2
            600;0;auxAvoir;;
            @active,@avoir,cat=v;
            %default
```

Dans ce cas, les valeurs par défaut deviennent : la catégorie syntaxique est *v*, la catégorie morphologique est *cat=v*, le cadre de sous-catégorisation est vide puisqu'il ne contient ni sujet, ni objet puisque l'entrée est entièrement figée, c'est pourquoi la redistribution par défaut est *%default*.

De même que pour les tables précédentes, le code de flexion se compose de plusieurs *inv* (pour chaque mot du sujet figé, un seul pour la table EC0 pour le sujet Ce, et deux pour les tables E01 et E0P1 correspondant à Det0 C0), puis d'autant de *0* que de clitiques figés avec le verbe, suivi de *v-être* puisqu'il s'agit du verbe *être*.

- En fait, à part la table EC0, pour les tables citées précédemment (C0, C0E, C0Q, CFF, E01 et E0P1), le problème est que certains mots du sujet peuvent être vides. Par exemple, si l'on considère le sujet Det0 C0 des tables E01 et E0P1, le Det0 peut être vide (<E>) ou être composé de plusieurs mots (par exemple, *le premier*), et le C0 peut également être composés de plusieurs mots (par exemple, *action de la justice*). Le nombre de mots constituant le sujet figé est donc variable, ce qui rend impossible la conversion au format Le*fff*, où il est nécessaire de repérer la position du verbe (pour indiquer sa table de flexion), afin de générer toutes les formes fléchies correspondantes. Il faudrait donc ajouter des colonnes catégorisant mieux chaque mot constituant le sujet figé.
  Notons également que lorsque le déterminant vaut *Det*, il peut valoir *un* ou *une* pour les noms au singulier mais pas *des* pour les noms au pluriel (par exemple, l'entrée *Det dépenses sont inconsidérées*) mais plutôt *les*, *ces*, *ses*, etc. Il faudrait donc définir la liste exacte des déterminants pour les remplacer dans le Le*fff* ;
- Pour la table 31I, même si l'on pourrait croire que le sujet est figé (C0 dans la construction de base C0 V W), il s'agit des verbes impersonnels, le sujet est donc *il*. Les valeurs par défaut deviennent ici : *v* pour la catégorie syntaxique, *cat=v* pour la catégorie morphologique, *%actif_impersonnel* pour la redistribution et le cadre de sous-cégorisation ne contient que le sujet avec la réalisation *sn*. Le code de flexion est calculé comme pour les verbes et correspond à *v-er:std* car tous les verbes appartiennent au premier groupe. Comme pour les tables de verbes, l'entrée peut contenir des adverbes tels que *Advm*, *jamais*, etc., il faut donc prendre en compte uniquement le premier mot de l'entrée, c'est-à-dire le verbe, puisqu'il sera fléchi.
  Dans la table, en réalité, le sujet *il* n'est pas accepté pour tous les verbes, donc cette information a été généralisée. En revanche, le pronom *ça* est accepté pour tous les verbes, ce qui nous a amené à ajouter la macros *@impers_ca* dans le Le*fff* pour pouvoir représenter cette information ;
- Pour les tables C0, C0E, C0Q, CFF, CPP, CPPN, CPPQ, CV et Z5D, le problème est plus important car il faudrait dédoubler chacune de ces tables. En effet, elles

comportent dans leur construction de base un + (par exemple, (Det1 C1+N1)), indiquant que l'argument est libre ou figé, ce qui ensuite dépend des entrées. Elles ne peuvent donc pas être traitées de la même manière et la construction de base doit être précisée pour chaque entrée.

Pour terminer, il reste encore un grand nombre de problèmes à résoudre, qui nécessitent une révision complète des tables avant d'obtenir une version des tables d'expressions figées au format Le*fff*. Notamment, il est nécessaire d'harmoniser les notations des codes de remplacement figurant dans les entrées, d'ajouter des colonnes dans certaines tables ayant un sujet figé afin de repérer la position du verbe et de dédoubler les tables acceptant en parallèle des arguments libres et figés.

8.2 Le lexique *LGLex*-Le*fff*

Le lexique issu de *LGLex* est appelé *LGLex*-Le*fff*, pour le différencier du Le*fff* lui-même. Le lexique des verbes simples obtenu contient 22 060 entrées pour 5 736 lemmes verbaux distincts[13], c'est-à-dire un nombre moyen d'entrées par lemme de 3,85. À titre de comparaison, le Le*fff* contient seulement 7 072 entrées verbales pour 6 818 lemmes verbaux distincts, soit un nombre moyen d'entrées par lemme de 1,04. Le lexique *LGLex*-Le*fff*, quoique décrivant moins de lemmes verbaux[14], est donc beaucoup plus couvrant en termes de constructions syntaxiques, et donc beaucoup plus ambigu. Les lemmes verbaux les plus ambigus dans le Le*fff* sont *tenir* et *(re)faire* (6 entrées), alors que dans le lexique *LGLex*-Le*fff* ,il s'agit des lemmes *tenir* (53 entrées), *jouer* (44 entrées) et *prendre* (35 entrées).

Au niveau extensionnel, le Le*fff* contient 361 268 entrées, alors que le lexique extrait de *LGLex* en contient 1 130 960.

Le lexique des noms prédicatifs obtenu contient 28 322 entrées pour 8 491 lemmes distincts (3,34 entrées/lemme). Le Le*fff* ne contient que 218 entrées de noms prédicatifs (1 entrée/lemme). Au niveau extensionnel, *LGLex*-Le*fff* contient 52 517 entrées pour les noms prédicatifs.

L'obtention de ce lexique à partir de *LGLex*, par le processus décrit ici est réalisée par *LGLex2ilex*, un script en Perl d'environ 1 500 lignes. La conversion proprement dite, c'est-à-dire l'exécution du script sur l'ensemble de *LGLex*, prend moins d'une minute. Si une nouvelle version des tables et de la table des classes est publiée, la construction du lexique correspondant au format Alexina est quasiment instantanée, et ne nécessite aucun nouveau développement.

Les détails nécessaires afin de générer le lexique *LGLex*-Le*fff* sont expliqués dans l'annexe D.

13. Le nombre de verbes distincts est identique à celui des tables (à 2 entrées près, cf. 8.1.4). En revanche, le nombre d'entrées ne correspond pas à celui des tables, puisque pour chaque entrée, les constructions trop éloignées de la CBME (non déductibles par une redistribution du Le*fff*) conduisent à des entrées secondaires avec le même identifiant (cf. cas 5 en 8.1.1).

14. Cela est dû aux verbes récents ou techniques présents dans le Le*fff*.

Quatrième partie

Résultats

9
Récapitulatif des travaux effectués

Dans ce chapitre, nous faisons un bilan sur l'avancement des travaux de conversion des tables dans la section 9.1, ainsi que sur l'évolution de la version 3 des tables dans la section 9.2. Enfin, nous montrons un extrait de la table 31R dans tous les formats disponibles actuellement dans la section 9.3.

9.1 Bilan sur l'avancement des travaux de conversion

Reprenons la Tab. 5.1 qui fait l'inventaire complet des tables (classées par catégorie) actuellement développées, pour y ajouter les différents formats auxquels elles ont été converties dans la Tab. 9.1 (Tolone, 2009). Pour chaque ensemble de tables nous indiquons leurs auteurs principaux (cf. la Tab. 5.1 pour les références associées), le nom des tables (cf. la Tab. 5.1 pour une liste plus détaillées, ainsi que le nombre de tables concernées), ainsi que les formats dans lesquels elles sont disponibles actuellement. Nous précisons si elles sont au format Excel, si elles sont incluses dans la table des classes appropriée (cf. section 5.4), si elles sont incluses dans le lexique *LGLex* (cf. chapitre 7), et enfin si elles sont incluses dans le lexique *LGLex*-Le*fff* (cf. chapitre 8).

9.2 Évolution de la version 3

Comme nous l'avons mentionné en 5.2.4, les tables disponibles ainsi que les différents lexiques auxquels elles ont été converties (*LGLex* et *LGLex*-Le*fff*) font partie de la version 3, qui a évoluée depuis septembre 2008. La version 3.3 est disponible depuis le 31 mars 2011 sur le site http://infolingu.univ-mlv.fr/ (Données Linguistiques >

Auteurs	Nom des tables	Format Excel	Table des classes	Lexique *LGLex*	Lexique *LGLex*-Le*fff*
(a) Verbes distributionnels (simples)					
M. Gross	1 à 18	OK	OK	OK	OK
J.-P. Boons, A. Guillet, C. Leclère (BGL)	32A ... 39	OK	OK	OK	OK
J.-P. Boons, A. Guillet, C. Leclère (BGL)	31H ... 35R	OK	OK	OK	OK
A. Guillet, C. Leclère (BGL)	36DT ... 38LR	OK	OK	OK	OK
C. Leclère, A. Borillo (BGL)	36S	OK	OK	OK	OK
C. Leclère (BGL)	35RR et 38RR	OK	OK	OK	OK
(b) Noms prédicatifs (simples et composés)					
J. Giry-Schneider	F1A à F91	OK	OK	OK	OK
J. Giry-Schneider	FN à FNPNN	OK	OK	OK	OK
A. Meunier	AN01 ... ANSY	OK	OK	OK	OK
G. Gross	AA ... IS2	OK	OK	OK	OK
J. Giry-Schneider,	AN07 à AN10	OK	OK	OK	OK
A. Balibar-Mrabti					
J. Giry-Schneider	ANDN et ANSN	OK	OK	OK	OK
D. de Négroni-Peyre	PSY et SYM	OK	OK	OK	OK
R. Vivès	APE1 à APP3	OK	OK	OK	OK
J. Labelle	ANA ... ANSU	OK	OK	OK	OK
L. Pivaut	FD1 à FD4	En cours	En cours	En cours	En cours
(c) Expressions figées (principalement verbales et adjectivales)					
J.-P. Boons, A. Guillet, C. Leclère (BGL)	31I	OK	OK	OK	En cours
M. Gross	A1 ... YA	OK	OK	OK	En cours
M. Gross	EAPC ... EPA	OK	OK	OK	En cours
M. Gross	EPAC à EPDETC	OK	OK	OK	En cours
M. Gross	E1 ... EDN	OK	OK	OK	En cours
J. Giry-Schneider	FC à FCPNN	OK	OK	OK	En cours
L. Danlos	Z à ZS	En cours	En cours	En cours	En cours
(d) Adverbes (simples et (semi-)figés)					
C. Molinier, F. Levrier	ADVMF à ADVPS	OK	OK	OK	/
M. Gross	PAC à PVCO	OK	OK	OK	/
(e) Adjectifs prédicatifs (simples)					
L. Picabia	ADJ1 à ADJ13	/	/	/	/
J. Giry-Schneider	ADJ01 à ADJSYM	En cours	/	/	/
(f) Adjectifs non prédicatifs (simples)					
E. Laporte	ADJLOCTABLE	OK	/	/	/
J. Giry-Schneider	DEDJA et DADJI	En cours	/	/	/

TABLE 9.1: Inventaire des tables du Lexique-Grammaire et avancement des travaux de conversion vers des formats électroniques exploitables

Lexique-Grammaire > Téléchargement), accompagnée des outils de conversion (*LGExtract* et *LGLex2ilex*). La Tab. 9.2 récapitule l'évolution de la version 3 [1].

Les catégories couvertes par les tables, au format CSV [2], et par le lexique *LGLex*, au format texte et XML, sont les verbes, les noms prédicatifs, les expressions figées et les adverbes. Des entrées supplémentaires de verbes, expressions figées et adjectifs prédicatifs ont été extraites de certaines tables (cf. section 6.6). De plus, les verbes et les noms prédicatifs sont au format *LGLex*-Le*fff* (intensionnel et extensionnel).

Parmi les informations sémantiques codées dans les tables, le format *LGLex*-Le*fff* garde celle qui est la plus complète, à savoir si un syntagme nominal peut être humain et/ou non humain (macros *@SujNhum*, *@ObjàN-hum*, etc.).

De plus, un README est disponible pour chacun des deux formats de lexique en français et en anglais (voir les annexes A et C), de même que pour chacun des deux outils de conversion (voir les annexes B et D), et enfin pour les tables afin de présenter leur format, leur codage, ainsi que celui de la table des classes, et d'y inclure toutes les références complètes. On spécifie également pour chacune des références, le nombre de tables concernées, ainsi que la liste complète des tables (cf. la Tab. 5.1).

Enfin, en plus des tables, nous avons également une table des classes par catégorie, ainsi que pour les verbes, la documentation de toutes les propriétés syntaxico-sémantiques en français et en anglais (voir l'annexe E), l'ordre de priorité dans le classement des verbes accompagné des formules définitoires de chaque table (voir l'annexe F) et l'arbre de classement (voir l'annexe G).

Les entrées verbales ont subit un grand nombre de modifications, notamment lors de la mise en correspondance des tables et de l'index éléctronique, tout en tenant compte des fiches cartonnées (cf. 5.5.2), puis de la conversion au format Le*fff* qui a permis la détection d'erreurs (cf. 8.1.4). Nous avons effectué une comparaison entre les entrées verbales de la première version de septembre 2008 (provenant de l'ensemble des tables, contrairement à ce qui a été diffusé) et celles de la version 3.3 de mars 2011. En 2008, nous avions 13 327 entrées verbales pour 5 714 verbes distincts (dans 59 tables [3]), ce qui totalise 540 nouvelles entrées verbales, dont 24 verbes distincts par rapport à la version 3.3. Parmi ces verbes distincts, présentés à la Tab. 9.3, on distingue 9 verbes supprimés (ayant été corrigés et correspondant pour la plupart à un verbe déjà existant), 8 verbes corrigés et 33 verbes ajoutés.

1. Signalons que le site `http://www-igm.univ-mlv.fr/~tolone/recherche.php` contient également la dernière version diffusée, ainsi que l'ensemble des versions précédentes de la version 3 présentées à la Tab. 9.2.

2. Le format CSV (voir `http://fr.wikipedia.org/wiki/Comma-separated_values`) permet de représenter les tables indépendamment du logiciel utilisé (Microsoft Excel, OpenOffice.org Calc, LibreOffice Calc, Gnumeric, etc.). Le logiciel doit cependant gérer l'Unicode, sinon des caractères incorrects peuvent apparaître. L'avantage est que c'est un format texte, il est donc facile de comparer différentes versions, ce qui n'est pas possible avec le format Excel.

3. Rappelons que les nouvelles tables sont : 2T, provenant de la division de la table 2 ; 32D, créée parallèlement à la table 32A ; 35LD, 35LS et 35LR, provenant de la division de la table 35L ; 38LHD, 38LHS et 38LHR, provenant de la division de la table 38LH.

Versions	Tables	*LGLex*	*LGLex*-Le*fff*
3.3 (31-03-2011)	Format CSV : – verbes (100%) – noms prédicatifs (100%) – expressions figées (100%) – adverbes (100%) – 4 tables des classes	Format texte/XML avec README (FR/EN) : – verbes (100%) – noms prédicatifs (100%) – expressions figées (100%) – adverbes (100%) Outil *LGExtract*	Format intensionnel/ extensionnel avec README (FR/EN) : – verbes (100%) – noms prédicatifs (100%) Outil *LGLex2ilex*
	Ajouts verbes : – ajout de propriétés – doc. propriétés – formules définitoires – arbre classement Ajouts adverbes : – propriétés paraphrases	Entrées suppl. : – verbes (100%) – expressions figées (100%) – adj. prédicatifs (100%)	
3.2 (30-06-2010)	Format Excel : – verbes (60%) – noms prédicatifs (60%) – expressions figées (60%) – adverbes (100%)	Format texte/XML avec README (FR/EN) : – verbes (60%) – noms prédicatifs (60%) – expressions figées (60%) – adverbes (100%)	Format intensionnel/ extensionnel avec README (FR/EN) : – verbes (60%) – noms prédicatifs (60%)
	Ajouts : – changements d'intitulés Ajouts verbes : – ajout d'entrées – codage de prépositions	Entrées suppl. : – verbes (60%) – expressions figées (80%) – adj. prédicatifs (100%)	Ajouts verbes : – macros *@SujNhum*, *@ObjàN-hum*, etc. = syntagme nominal humain et/ou non humain
3.1 (27-11-2009)	Format Excel : – verbes (60%) – noms prédicatifs (60%)	Format texte/XML avec README (FR/EN) : – verbes (60%) – noms prédicatifs (60%)	Format intensionnel : – verbes(60%)
3.0 (18-12-2008)	/	Format texte/XML : – verbes (60%) – noms prédicatifs (60%)	/
3.alpha (08-09-2008)	/	Format XML : – verbes (60%) – noms prédicatifs (60%)	/

TABLE 9.2: Évolution de la version 3

Verbes supprimés	Verbes corrigés	Verbes ajoutés
bailler	balloter → ballotter	affrioler
cafeter	fossoyeur → fossoyer	aponter
charge	godeliser → gödeliser	approximer
énupler	libéller → libeller	attitrer
fiche	pied££au-cuter → pied-au-cuter	avérer
fouire	sous-payer → sous-tendre	barbifier
qoranniser	théatrâliser → théâtraliser	bicher
rencontre	vert££de-griser → vert-de-griser	coraniser
sous-employer		courser
		cuveler
		dégotter
		désaccoupler
		droper
		ébaubir
		enchrister
		gasconner
		infibuler
		insécuriser
		institutionnaliser
		iodler
		molarder
		narrer
		n-upler
		raper
		raplatir
		rassurer
		reconstruire
		renauder
		rossignoler
		rouscailler
		skier
		ventriloquer
		zieuter

Table 9.3: Comparaison entre les verbes distincts de la première version de 2008 avec ceux de la version 3.3

9.3 Extraits de la table 31R dans tous les formats disponibles

9.3.1 Table en version 1, 2 et 3

La Fig. 9.1 représente un extrait de la table 31R en version 1, telle qu'elle est publiée dans la littérature (Boons *et al.*, 1976b). Comme déjà exliqué en 5.2.4, on peut y voir des cartouches horizontaux, qui regroupent une famille de propriétés, ainsi que des dépendances entre propriétés. En, effet, les six premières colonnes sont des propriétés distributionnelles de l'argument N0, comme l'indique le cartouche horizontal intitulé N0. De plus, l'intitulé N0 =: Npc (où Npc dénote le corps ou une partie du corps d'une personne) dépend de l'intitulé N0 =: Nhum, ce qui est représenté par le deuxième cartouche vertical inclus dans le premier. Notons que la construction de base décrite dans (Boons *et al.*, 1976b) (p. 262) est N0 V.

La table 31R en version 2 n'est pas disponible sur le site http://infolingu.univ-mlv.fr (Données Linguistiques > Lexique-Grammaire > Visualisation) car elle ne fait pas partie des 60% librement distribué.

La table 31R en version 3 (actuelle), disponible sur le site, est montrée à la Tab. 9.4 (avec l'ensemble de ses propriétés) pour les deux verbes *croupir* et *croustiller*, présents à Fig. 9.1 et pour le verbe *croûter* qui a été ajouté à la table depuis et qui n'a pas encore été codé (codage ∼). Remarquons que le verbe *cuver* n'est plus dans la table 31R actuelle, il figure dans la table 32R2 (*Max cuve son vin*) et 38LR (*On a cuvé le moût dans la cuve*). On peut voir également que la propriété N actif V N0 a été ajoutée dans la table sans avoir été codée, ce qui explique son codage ∼.

Les colonnes Ppv =: se figé, Ppv =: en figé, Ppv =: y figé, Ppv =: Neg et Neg ont été ajoutée à la version 3 avec un codage binaire, codage qui doit concorder avec les informations contenues dans la colonne <ENT>Ppv, laquelle n'est gardée que pour des raisons de lisibilité (cf. section 3.2). Les colonnes Aux =: avoir, Aux =: être et N0 être V-n (forme V-n) ont été ajoutées lors de la réintégration du codage de certaines informations concernant les auxiliaires, les V-n et la traduction en anglais des entrées (<ENGLISH>) dans les tables de verbes (Paumier, 2003). L'intitulé il V N0 W ayant la même signification que l'intitulé [extrap] présent dans les tables de M. Gross, la colonne a été harmonisée (cf. 5.3.2). Enfin, certaines propriétés concernant les parties du corps ont été supprimées des tables car elles étaient trop productives et non spécifiques à quelques classes particulières. De plus, elles sont ignorées lors de la conversion au format *LGLex*, tout comme les propriétés concernant les V-n (cf. section 7.3).

Il faut ajouter à cette table le codage de la table des classe à la Fig. 9.5, avec les 33 propriétés que nous avons jugées pertinentes pour la table 31R sur les 551 propriétés des tables verbales (le codage des propriétés supprimées étant toutes codées − ou ?). Cela explique par exemple la disparition de la colonne N0 =: N-hum dans la table, qui est codée + dans la table des classes, puisque son codage est + pour l'ensemble des entrées. De même, cela permet à la construction de base N0 V d'être explicitement codée. Toutes

N_0 = N hum	N_0 = N pc	N_0 = N -hum	N_0 = N nr	N_0 = V Ω	N_0 = V-n		N_0 est V-ant	N_0 est V pp	N^0_{pc} lui V	N_0 V de N^0_{pc}	il V N_0 Ω
-	+	+	-	-	-	croupir	+	+	-	-	+
-	-	+	-	-	-	croustiller	+	-	+	-	+
-	-	+	-	-	-	cuver	-	+	-	-	+

Figure 9.1: Extrait de la table 31R des verbes distributionnels (version 1)

N0 =: Nhum	N0 =: Nnr	N0 =: V-inf W	N0 être V-n	<ENT>Ppv	Ppv =: se figé	Ppv =: en figé	Ppv =: y figé	Ppv =: Neg	**<ENT>V**	Neg	Aux =: avoir	Aux =: être	N0 être V-ant	N0 être Vpp	N0 V de N0pc	[extrap]	N actif V N0	N0 être V-n (forme V-n)	<OPT>
−	−	−	−	<E>	−	−	−	−	*croupir*	−	+	−	+	+	−	+	∼	*<E>*	*L'eau croupit*
−	−	−	−	<E>	−	−	−	−	*croustiller*	−	+	−	+	−	−	+	∼	*<E>*	*Le gâteau croustille*
∼	∼	∼	∼	<E>	−	−	−	−	*croûter*	−	+	−	∼	∼	∼	∼	∼	*<E>*	*Le savon croûte*

Table 9.4: Extrait de la table 31R des verbes distributionnels (version 3)

les colonnes présentes dans la table 31R sont codées o dans la table des classes, puisque leur valeur varie en fonction des entrées. Certaines colonnes ont été codées – comme par exemple, le passif ou la distribution Nnc du N0. D'autres doivent être codées (codage O), comme la traduction en anglais de chaque entrée (<ENGLISH>). Enfin, d'autres ont un codage qui reste à déterminer (codage ?), par exemple la construction N0 être V-eux W.

9.3.2 Lexique *LGLex*

Voici l'extrait du lexique *LGLex* correspondant aux entrées verbales *croupir*, *croustiller* et *croûter* de la table 31R. On peut voir que l'entrée *croûter* (ayant le statut **to encode**) n'est pas complète car elle n'est pas codée dans la table, mais elle contient tout de même les informations codées dans la table des classes, à savoir sa construction de base N0 V et la propriété définitoire N0 =: N-hum. Les deux autres entrées (ayant le statut **to complete**) sont codées dans la table, à part pour la propriété N actif V N0, qui n'est codée pour aucune entrée (codage ∼) :

```
ID=V_31R_77;status=to complete
lexical-info=[cat="verb",verb=[lemma="croupir"],
              aux-list=(avoir="true"),prepositions=(),locatifs=()]
 args=(const=[pos="0",
```

Propriété \ table	V_31R
[extrap]	o
[passif de]	−
[passif par]	−
<ENGLISH>	O
<ENT>Ppv	o
<ENT>V	o
<OPT>	o
Aux =: avoir	o
Aux =: être	o
N0 =: chemin	?
N0 =: Nhum	o
N0 =: N-hum	+
N0 =: N-hum métaphore	?
N0 =: Nnc	−
N0 =: Nnr	o
N0 =: Qu P	?
N0 =: V-inf W	o
N0 être V-ant	o
N0 être V-eux W	?
N0 être V-n	o
N0 être V-n (forme V-n)	o
N0 être Vpp	o
N0 V	+
N0 V de N0pc	o
N actif V N0	o
Neg	o
Ppv =: en figé	o
Ppv =: la figé	−
Ppv =: le figé	−
Ppv =: les figé	−
Ppv =: Neg	o
Ppv =: se figé	o
Ppv =: y figé	o

Table 9.5: Extrait de la table des classes verbes distributionnels

```
             dist=(comp=[cat="NP",nothum="true",
                   introd-prep=(),introd-loc=(),
                   origin=(orig="N0 =: N-hum")])])
 all-constructions=[absolute=(construction="true::N0 V",
                             construction="o::N0 être V-ant",
                             construction="o::N0 être Vpp"),
                   relative=(construction="[extrap]")]
 example=[example="L'eau croupit"]

ID=V_31R_78;status=to complete
lexical-info=[cat="verb",verb=[lemma="croustiller"],
              aux-list=(avoir="true"),prepositions=(),locatifs=()]
args=(const=[pos="0",
             dist=(comp=[cat="NP",nothum="true",
                   introd-prep=(),introd-loc=(),
                   origin=(orig="N0 =: N-hum")])])
all-constructions=[absolute=(construction="true::N0 V",
                            construction="o::N0 être V-ant"),
                   relative=(construction="[extrap]")]
example=[example="Le gâteau croustille"]

ID=V_31R_79;status=to encode
lexical-info=[cat="verb",verb=[lemma="croûter"],
              aux-list=(avoir="true"),prepositions=(),locatifs=()]
args=(const=[pos="0",
             dist=(comp=[cat="NP",nothum="true",
                   introd-prep=(),introd-loc=(),
                   origin=(orig="N0 =: N-hum")])])
all-constructions=[absolute=(construction="true::N0 V"),
                   relative=()]
example=[example="Le savon croûte"]
```

9.3.3 Lexique *LGLex*-Le*fff*

Voici les entrées verbales *croupir*, *croustiller* et *croûter* de la table 31R au format Le*fff*. On peut voir que les deux entrées des verbes *croupir* et *croustiller* sont identiques alors que dans les tables, la construction N0 être Vpp les différencie. Cette propriété n'est en effet pas prise en compte lors de la conversion. En revanche, on voit que l'entrée du verbe *croûter* ressemble également aux autres, mais n'a pas la redistribution *%actif_impersonnel*, la colonne [extrap] n'étant pas codée (codage ∼) pour cette entrée :

```
croupir___V_31R_76 v-ir2
                   100;Lemma;v;
                   <Suj:cln|sn>
                   cat=v;@SujN-hum;
                   %actif,%actif_impersonnel

croustiller___V_31R_77 v-er:std
                       100;Lemma;v;
                       <Suj:cln|sn>
```

```
                        cat=v;@SujN-hum;
                        %actif,%actif_impersonnel

croûter___V_31R_78 v-er:std
                   100;Lemma;v;
                   <Suj:cln|sn>
                   cat=v;@SujN-hum;
                   %actif
```

10
Intégration dans FRMG et évaluation

La section 10.1 montre comment le lexique *LGLex*-Le*fff* a été couplé avec l'analyseur syntaxique à grande échelle FRMG (cf. section 4.2), comme nous l'avons expliqué dans Sagot et Tolone (2009b); Tolone et Sagot (2011), en expliquant également les raisons du choix de l'analyseur. La section 10.2 évalue cet analyseur sur le corpus de référence de la première campagne d'évaluation Passage de 2007, en comparant sa version basée sur le Le*fff* avec notre version reposant les entrées des verbes simples et des noms prédicatifs des tables du Lexique-Grammaire converties, comme nous l'avons initialement fait pour les verbes sur le corpus de référence de la campagne EASy (Sagot et Tolone, 2009a; Tolone et Sagot, 2011). La section 10.3 présente la fouille d'erreurs effectuée sur le corpus EASy. La section 10.4 présente l'évaluation manuelle réalisée pour valider la qualité des analyses, avant de conclure dans la section 10.5.

10.1 Intégration dans l'analyseur syntaxique FRMG

Nous avons réalisé l'intégration des tables dans l'analyseur FRMG, afin d'évaluer leur impact lors d'une l'analyse. Les raisons qui nous ont poussée à choisir cet analyseur sont les suivantes :

– sa disponibilité, contrairement à Cordial de l'entreprise Synapse, le premier système de la campagne d'évaluation Passage de 2007 (cf. chapitre 2) ;
– son formalisme TAG, qui est lexicalisé et traite les mots composés, et qui a donc besoin d'un lexique à large couverture ;
– ses performances, puisque lors de la campagne Passage de 2007, il est troisième sur les dépendances (dans le groupe au-dessus de 60%) ;
– son amélioration entre la campagne EASy de 2004 et Passage de 2007, en particulier sur la désambiguïsation et la conversion vers le format de la campagne, puisqu'il

passe de 69% à 89% pour les constituants et de 41% à 63% pour les dépendances. De plus, depuis la campagne de 2007, il s'est encore amélioré de 6% pour les constituants et de 7% pour les dépendances, notamment par l'augmentation du nombre d'analyses complètes ;
– son utilisation par une large communauté et la bonne interaction entre les utilisateurs et le développeur ;
– son lexique, le Le*fff*, qui est un lexique à large couverture, utilisé également par une large communauté.

L'intégration du lexique *LGLex*-Le*fff* dans l'analyseur FRMG est immédiate : FRMG *Lexer* fait appel à une base de données lexicales construite à partir du Le*fff*. Il suffit de remplacer les entrées des verbes simples du Le*fff* par celles du lexique *LGLex*-Le*fff*, d'ajouter les entrées des noms prédicatifs du lexique *LGLex*-Le*fff*, absentes du Le*fff*, et de conserver les autres entrées du Le*fff*. Après avoir construit la base de données lexicales correspondante, il faut seulement spécifier à FRMG d'utiliser cette dernière. Le résultat est une variante de l'analyseur FRMG, que nous notons FRMG_{LGLex}, par opposition à la variante standard notée $\text{FRMG}_{\text{Le}fff}$.

FRMG étant en constante amélioration, l'intégration du lexique *LGLex*-Le*fff* a permis de déceler quelques problèmes qui ont été corrigés par Éric de la Clergerie. Les modifications ont eu lieu à divers niveaux :
– dans la méta-grammaire FRMG : gestion optionnel/non optionnel ;
– dans FRMG *Lexer* : résolution des problèmes de réalisation des arguments ;
– dans le Le*fff* : ajout de nouvelles prépositions codées dans les tables de verbes (comme *chez* pour le verbe *susciter*).

10.2 Évaluation et discussion

Nous avons évalué $\text{FRMG}_{\text{Le}fff}$ et FRMG_{LGLex} en analysant la partie annotée manuellement pour la première campagne Passage de 2007, composée de 4 706 phrases du corpus EASy de styles variés (cf. 2.1.2). Avant de discuter des résultats de cette expérience, certaines précautions sont à prendre :
– le processus de conversion décrit ici et son implémentation contiennent certainement des erreurs, de plus, nous évaluons FRMG lorsqu'il utilise les entrées converties à partir des tables, et non pas les entrées telles qu'elles sont dans les tables ;
– la majorité des noms prédicatifs n'ont pas pu être évalués car FRMG ne prend que ceux n'ayant pas de déterminant ;
– la campagne Passage ne permet pas d'évaluer toutes les informations contenues dans les tables (informations sémantiques par exemple)
– le Le*fff* a été développé en parallèle aux campagnes EASy et Passage, contrairement aux tables ; certains choix faits dans le guide des annotations syntaxiques de référence PEAS [1] ont pu influencer certains choix faits dans le développement du Le*fff*, alors que ce n'est évidemment pas le cas pour les tables ;

1. http://www.limsi.fr/Individu/anne/Guide/PEAS_reference_annotations_v2.2.html

– *LGLex* a été complété par diverses entrées verbales venant du Le*fff*, qui ne font pas partie du lexique *LGLex* : entrées pour les auxiliaires et semi-auxiliaires, certains verbes à montée, les verbes impersonnels et les entrées pour les têtes syntaxiques des constructions à verbe support ; il se peut que d'autres entrées soient encore à rajouter. De plus, toutes les entrées verbales non codées dans les tables (codage ∼) manquent également.

Les résultats comparatifs sur les constituants et les relations (dépendances entre mots pleins) entre ces deux analyseurs sont donnés à la table 10.1 [2], avec le détail pour quelques sous-corpus illustratifs. Les résultats sont donc pour l'instant légèrement meilleurs pour FRMG$_{\mathrm{Le}\mathit{fff}}$, mais également bons pour FRMG$_{LGLex}$. En effet, une f-mesure pour les relations au-dessus de 60% est meilleure que les résultats de FRMG avec le Le*fff* lors de la campagne de 2007 (59,65% de f-mesure pour 56% de couverture). Néanmoins, les derniers points de f-mesure découle d'une adaptation plus fine entre la grammaire et le lexique, et de la recherche des erreurs ou incomplétudes lexicales. Il est en effet normal que tout lexique possède des entrées erronées qu'il est difficile de trouver.

De plus, même si les résultats ne sont pas meilleurs, nous ne pensons pas que cela remette en question la pertinence de l'utilisation des tables du Lexique-Grammaire en analyse syntaxique, notamment au vu des précautions ci-dessus. En particulier, nous restons convaincue que l'utilisation d'une ressource lexicale aussi riche que possible reste un moyen efficace pour améliorer la qualité d'un analyseur syntaxique, comme l'ont montré par exemple la mise en œuvre des travaux décrits dans Sagot et de La Clergerie (2008). On peut toutefois constater que les temps d'analyse sont plus deux fois plus élevés avec FRMG$_{LGLex}$ qu'avec FRMG$_{\mathrm{Le}\mathit{fff}}$ (temps médian par phrase de 0,62 s contre 0,26 s), ce qui provient certainement du nombre d'entrées par lemme qui est trois plus élevé dans *LGLex* que dans le Le*fff*, comme nous l'avons mentionné dans la section 8.2. Du reste, ce temps d'analyse plus élevé conduit nécessairement à un plus grand nombre d'échecs d'analyse par dépassement du délai maximum autorisé, ce qui conduit à la construction d'analyses partielles, nécessairement de moins bonne qualité.

Sur deux sous-corpus, c'est toutefois FRMG$_{LGLex}$ qui obtient les meilleures évaluations en constituants. De même, les résultats sur les relations sont meilleurs avec FRMG$_{LGLex}$ sur un sous-corpus. L'analyse des résultats montre les faits suivants :

– FRMG$_{LGLex}$ donne de meilleurs résultats que FRMG$_{\mathrm{Le}\mathit{fff}}$ pour certaines relations, comme modifieur d'adjectif (MOD-A) et modifieur d'adverbe (MOD-R), mais également pour deux relations pour lesquelles les résultats sont mauvais d'un côté comme de l'autre : modifieur de préposition (MOD-P) et apposition (APP) ;
– la relation attribut du sujet ou de l'objet (ATB-SO) est celle pour laquelle la différence en rappel est la plus importante (34,0% contre 58,4%) ;
– le degré d'ambiguïté lexicale, bien plus élevé dans FRMG$_{LGLex}$ que dans FRMG$_{\mathrm{Le}\mathit{fff}}$, conduit à un taux d'ambiguïté plus grand au niveau de l'analyseur, et donc d'autant

2. Ces mesures sont calculées avec les outils d'évaluation propres à l'INRIA, qui donnent généralement des valeurs plus faibles de quelques points que les résultats officiels. En effet, les mesures des campagnes EASy et Passage sont un peu plus laxistes, par exemple en autorisant des variations sur les limites des constituants. Enfin, il est à noter que les données de références ont été en partie corrigées au cours des années.

	Constituants		Relations	
Sous-corpus	$\text{FRMG}_{\text{Le}fff}$	FRMG_{LGLex}	$\text{FRMG}_{\text{Le}fff}$	FRMG_{LGLex}
`general_lemonde`	**88.22%**	84.60%	**62.73%**	59.01%
`litteraire_2`	**88.91%**	88.46%	**65.28%**	62.43%
`mail_9`	**82.60%**	81.90%	**58.55%**	56.00%
`medical_3`	85.04%	**85.89%**	64.79%	**65.26%**
`oral_delic_4`	78.80%	**81.79%**	**51.67%**	51.14%
`questions_amaryllis`	**91.30%**	90.73%	**66.56%**	64.77%
total	***87.05%***	*85.53%*	***63.10%***	*60.25%*

TABLE 10.1: Résultats comparatifs de $\text{FRMG}_{\text{Le}fff}$ et FRMG_{LGLex} sur le corpus de référence de la première campagne Passage, exprimé en terme de f-mesure.

plus de risque de se tromper au moment de la désambiguïsation. En effet, comme vu en 4.2.3, le désambiguïsateur utilisé par FRMG est à base de règles heuristiques pondérées[3].

Ainsi, le nombre élevé d'arguments verbaux listés dans les cadres de sous-catégorisation de *LGLex* conduit à induire en erreur l'heuristique de désambiguïsation habituelle selon laquelle «on préfère les arguments aux modifieurs» : tout syntagme pouvant être analysé comme un argument verbal a tendance à l'être. Dans une phrase comme *[...] on estime que cette décision [ferait] dérailler le processus de paix*, FRMG_{LGLex} fait de *de paix* un argument de *estimer* (*estimer qqch de qqn*), là où $\text{FRMG}_{\text{Le}fff}$ ne se trompe pas. Notons que si les traits sémantiques étaient pris en compte dans l'analyseur, *LGLex* ne conduirait pas à une erreur, puisque la table 6 fournit la construction N0 V N1 de N2hum et le substantif *paix* n'est pas humain, trait sémantique qui devrait également figurer dans le lexique des substantifs.

On peut faire le constat suivant : de nombreuses phrases reçoivent une analyse complète par FRMG_{LGLex} alors que ce n'est pas le cas par $\text{FRMG}_{\text{Le}fff}$, et inversement. Par exemple, sur le sous-corpus `general_lemonde`, 177 phrases sont entièrement reconnues par les deux analyseurs, 85 seulement par $\text{FRMG}_{\text{Le}fff}$, 76 seulement par FRMG_{LGLex}, et 111 par aucun des deux. L'expérience montre que la qualité des résultats est très supérieure, de plus de 10 points en moyenne, sur les phrases analysées complètement par rapport à celles qui reçoivent plusieurs analyses partielles. On peut donc envisager de construire un système très simple qui analyserait une phrase avec chacun des deux analyseurs, et qui, dans les cas où seulement l'un des deux propose une analyse complète, ne garde que celle-ci (ce qu'il faut faire dans les autres cas reste à étudier). Les résultats sont susceptibles d'être meilleurs que l'usage exclusif de l'un ou l'autre.

Il importe également de bénéficier de cette complémentarité entre les deux ressources. Il est donc intéressant d'étudier les différences entre les erreurs faites par chacun de ces deux analyseurs, y compris au moyen de techniques automatiques (Sagot et de La Cler-

3. On peut penser qu'un modèle statistique de désambiguïsation pourrait mieux gérer l'ambiguïté provenant de *LGLex*. Nous ne sommes pas en mesure de confirmer ou d'infirmer empiriquement cette hypothèse.

gerie, 2008), comme nous allons le voir à présent. Le but étant d'améliorer les différentes ressources, voire de détecter automatiquement des erreurs dans les lexiques.

10.3 Fouille d'erreurs

Nous avons effectué une fouille d'erreurs dans les sorties de l'analyseur syntaxique FRMG sur les verbes du corpus EASy[4]. Elle reprend les principes de fouilles d'erreurs présentés dans Sagot et de La Clergerie (2008), tout en l'adaptant. En effet, dans ce travail, l'objectif était en première approximation d'identifier les formes dont la présence dans une phrase tend à rendre la phrase inanalysable[5]. Pour cette expérience, cet objectif est adaptée afin d'identifier les entrées d'un lexique **hyp** qui semblent dégrader les performances de FRMG comparativement à un lexique de référence **ref**, ici le Le*fff* : l'idée est de trouver les formes, et plus précisément les formes verbales, dont la présence dans une phrase *analysable avec le lexique de référence* **ref** tend à rendre cette phrase *inanalysable avec le lexique* **hyp**.

Le corpus EASy est trop petit pour obtenir des statistiques suffisantes pour un diagnostic complet des ressources. Mais l'algorithme ne nécessitant pas de disposer d'un corpus de référence (seule l'analysabilité, c'est-à-dire la couverture, étant exploitée), nous pourrons dans un avenir proche l'utiliser sur un corpus bien plus volumineux (Tolone *et al.*, 2011).

Les principaux suspects dans *LGLex*, si l'on regarde en détail les 5 premiers verbes présents dans les 12 phrases qui n'ont pas été analysées, sont :

- *s'accroître* dans *la population s' accroît plus rapidement dans les régions les plus pauvres du monde*, car la forme pronominale est manquante ;
- *ressaisir* dans *Tout ce passé me ressaisit avec une force extrême*, ou dans *S' écria M Levrault tombant en arrêt et déjà prêt à ressaisir sa proie*, car la forme non pronominale est manquante ;
- *soupçonner* dans *il ne devinait rien et ne soupçonnait pas qu' il pût y avoir quelque anguille sous roche*, car le verbe n'est pas codé avec une complétive comme premier complément : l'entrée appropriée est celle de la table 6 (illustrée par l'exemple *Max soupçonne qu'il va pleuvoir*), mais fait partie des entrées qui ne sont pas codées (codage ∼) ;
- *impliquer* dans *Son évaluation implique d' inclure ces patients dans des essais thérapeutiques*, car il y a une erreur de conversion au format Le*fff* : cette entrée vient de la table 10 (*Ceci implique pour Max que Léa le quittera*) qui accepte comme construction de base N0 V N1 Prép N2, mais également la construction N0 V N1 et la distribution N1 = : de V2-inf W, ce qui est traduit au format Le*fff* par 2 entrées avec les cadres de sous-catégorisation suivants :

4. Nous avons réalisé ce travail en collaboration avec Eric de La Clergerie en 2011.

5. Une forme suspecte doit aussi avoir tendance à apparaître seule dans de telles phrases ou en cooccurrence avec des formes qui ne sont pas (trop) suspectes à l'échelle du corpus entier. On trouvera le détail du modèle sous-jacent dans Sagot et de La Clergerie (2008) ; il s'exprime sous forme d'une paire d'équations mutuellement récursives reflétant le niveau local des phrases et le niveau global du corpus. L'algorithme de résolution est un algorithme de point fixe.

```
<Suj:cln|scompl|sn|de-sinf,Obj:sn|scompl|de-sinf|cla,Obl:pour-sn|sn>
<Suj:cln|scompl|sn,Obj:sn|scompl|cla>
```

au lieu d'avoir un seul même cadre avec le deuxième complément facultatif.

Les autres suspects dans *LGLex* sont seulement des erreurs ou des phrases complexes :
- *monder* qui n'est pas un verbe dans 3 phrases semblables à *Bonjour tout_le_monde* ;
- *accroître* dans une phrase sans sujet : - *Accroître leurs parts de marché* ;
- *accroître* employé comme adjectif dans *les promesses d' autonomie accrue* ;
- *soupçonner* dans *Celui qui ne ferait qu'errer autour de la royale abbaye, n'en soupçonnerait nullement la beauté réelle*, car le sujet est complexe ;
- *impliquer* employé dans la phrase complexe : *Tous les agonistes dopaminergiques stimulent les récepteurs D2, initialement considérés comme étant seuls impliqués dans les effets moteurs de la dopamine.*

Les techniques de fouille d'erreurs permettant de comparer (dans un sens ou l'autre) les verbes de deux lexiques sont un moyen de repérer plus rapidement ces entrées. Il importe donc de refaire cette expérience sur un corpus plus volumineux afin de détecter le plus grand nombre de verbes manquants dans les tables ou y figurant sans être codés (codage ∼), afin de compléter les tables ; mais également de trouver les constructions manquantes pour certains verbes. Nous nous attendons à ce que la plupart des constructions manquantes proviennent du processus de conversion, mais certaines proviendront peut-être d'erreurs se situant dans les tables du Lexique-Grammaire. De telles considérations permettrait d'améliorer ces dernières.

10.4 Évaluation manuelle

Même si les résultats sont bons si l'on se compare à l'état de l'art, il importe d'évaluer la qualité des analyses. Pour cela nous avons vérifié manuellement, pour quelques phrases extraites du corpus de référence de la première campagne Passage, si les entrées trouvées lors de l'analyse avec FRMG correspondent à celles attendues d'après la classification du Lexique-Grammaire. Bien entendu, cela ne dépend pas que du contenu dans les tables mais également du fonctionnement de FRMG, notamment de son module de désambiguïsation, ce qui oblige à être prudent envers les résultats (en plus des précautions déjà citées) :
- différentes entrées peuvent avoir la même construction et ne sont donc pas détectables ;
- les compléments circonstanciel ne sont pas décrits dans les tables, donc tout complément peut être pris pour un complément circonstanciel et donc être accepté même s'il ne figure pas dans le cadre de sous-catégorisation de l'entrée ;
- nous évaluons après désambiguïsation, alors que dans certains cas la bonne table figure en mode non désambiguïsé, parmi un grand nombre d'autres tables, ce qui résulte plus difficile à évaluer. Le module de désambiguïsation de FRMG basé sur des heuristiques ne donne pas forcément la priorité aux bonnes tables.

Notons que les tables de noms prédicatifs ne font pas partie de cette évaluation puisque les seuls noms prédicatifs pris en compte par FRMG sont ceux n'ayant pas de

déterminant, et ils sont rares dans les corpus.

Nous avons extrait les premières phrases de deux sous-corpus (`general_lemonde` et `general_elda`), en ne gardant que celles ayant donné une analyse avec FRMG (en mode désambiguïsé) et contenant au moins un verbe[6]. Nous avons indiqué en gras les identifiants des verbes trouvés dans le lexique *LGLex*, en sachant que les verbes trouvés grâce au Le*fff* n'ont pas été répertoriés (il s'agit notamment des auxiliaires *être* et *avoir*). Nous avons regroupé les assignations correctes d'entrées codées dans les tables, les assignations incorrectes d'entrées codées dans les tables et les assignations incorrectes d'entrées non codées dans les tables. Notre expérience c'est limitée aux 33 premiers verbes de chaque sous-corpus (sans compter ceux qui ont été identifiés comme verbe mais qui n'en sont pas) et a été validée par des linguistes différents pour chacun des sous-corpus.

La Tab. 10.2 contient les 21 premières phrases du sous-corpus `general_lemonde` ayant donné une analyse avec FRMG. Ces phrases contiennent 33 verbes[7] reconnus grâce aux tables. Nous donnons aux Tab. 10.3 à 10.5 les résultats validés par Eric Laporte.

Au total, 48% des assignations d'entrées sont correctes pour le sous-corpus `general_lemonde`, soit 59% si l'on ne tient pas compte des entrées non codées dans les tables.

La Tab. 10.6 contient les 14 premières phrases du sous-corpus `general_elda` ayant donné une analyse avec FRMG. Ces phrases contiennent 33 verbes[8] reconnus grâce aux tables. Nous donnons aux Tab. 10.7 à 10.9 les résultats validés par Christian Leclère et Takuya Namakura.

Au total, 33% des assignations d'entrées sont correctes pour le sous-corpus `general_elda`, soit 42% si l'on ne tient pas compte des entrées non codées dans les tables.

Nous montrons à la Tab. 10.10 comment le taux de succès dans les deux sous-corpus varie en fonction du nombre d'entrées de chaque verbe dans la classification du Lexique-Grammaire. Par exemple, le verbe *utiliser* ne contient qu'une seule entrée alors que le verbe *être* en contient 27, ce qui produit beaucoup plus d'ambiguïtés.

Le faible nombre de phrases évaluées manuellement ne permet pas d'en conclure davantage. Cette évaluation pourrait être en effet poursuivie afin d'obtenir des résultats plus représentatifs. Malheureusement, cette validation n'est possible que par des linguistes connaissant bien les tables du Lexique-Grammaire.

6. Notons que dans le sous-corpus `general_elda`, une phrase peut être composée d'une listes à tirets avec plusieurs phrases. FRMG analyse uniquement la première phrase correspondant au premier tiret, puisqu'il s'agit d'une analyse phrase par phrase. Pour plus de lisibilité, nous n'avons gardé que cette première phrase dans les listes à tirets, les autres n'ayant pas été analysées.

7. En réalité, 35 verbes, mais nous mettons de côté l'adjectif *mesurées* (phrase 159), ainsi que le nom *président* (phrase 7.1).

8. En réalité, 34 verbes, mais nous mettons de côté l'adjectif *appropriées* (phrase 48).

1 *Depuis quelques semaines , les rapports entre les deux camps se dégradent___***V_32PL_46__1** *.*
2 *Le gouvernement intérimaire a visiblement décidé___***V_8_41__1** *d' asphyxier___***V_32H_48__1** *économiquement le " Taylorland " , en imposant___***V_37M1_111__1** *un embargo total sur les marchandises à destination des zones sous contrôle du FNPL et en remplaçant___***V_32H_592__1** *les coupures de 5 dollars libériens , les plus utilisés___***V_11_225__1** *, par de nouveaux billets .*
4 *Pendant ce temps , le drame des Libériens se poursuit___***V_37M1_159__1** *.*
7.1 *" Vous venez___***V_2_190__1** *nourrir___***V_37M1_140__2** *les Kranhs " (les membres de l' ethnie du défunt président___***V_32R3_692__1** *Samuel Doe) lui a lancé___***V_9_236__1** *le petit seigneur de guerre qui régnait___***V_35R_372__1** *sur son " check point " , son barrage routier .*
8 *Dans cette région , les enfants , les cheveux décolorés et le ventre ballonné , souffrent___***V_35R_438__1** *de kwashiorkor .*
9 *Ils manquent___***V_34L0_61__1** *de protéines .*
54 *Boutros Boutros-Ghali , effectuera___***V_32R2_170__1** *son premier voyage ès qualités à l' étranger en se rendant___***V_2_146___1** *le 10 janvier à Paris , puis le 13 janvier à Londres .*
55 *Ce premier déplacement n' a___***V_38R_31__1** *rien d' officiel , a précisé___***V_9_295__1** *son porte-parole , en ajoutant___***V_10_9__1** *qu' il ne s' agissait que d' une visite de travail .*
56 *La France a été___***V_33_60__1** *l' un des principaux artisans au sein du Conseil de sécurité de l' élection de M .*
57 *Boutros-Ghali , et Londres préside___***V_32R3_692__1** *pendant le mois de janvier le Conseil de sécurité . _*
58 *En France , 19 % des femmes actives ont été victimes de harcèlement sexuel sur leur lieu de travail .*
59 *C' est___***V_5_63__1** *ce que révèle_***V_32H_610__1** *un sondage rendu___***V_39_79___1** *public , mardi 7 janvier , par Mme Véronique Neiertz , secrétaire d' Etat aux droits des femmes et à la vie quotidienne .*
62 *Les victimes se plaignent___***V_13_111__2** *en premier de " propos et gestes douteux " (63 %) , mais aussi d' " avances répétées malgré un refus " (60 %) .*
63 *Dans 48 % des cas , elles parlent___***V_5_93__1** *d' " un climat d' ensemble déplaisant " , et 12 % des avances sont assorties d' un chantage .*
79 *Il s' élève_***38LD_100__1** *à 9 534 .*
81 *L' effort de sécurité routière ne s' en poursuit___***V_6_346__1** *pas moins .*
83 *Georges Sarre , secrétaire d' Etat chargé___***V_37M6_63__1** *des transports routiers et fluviaux .*
157 *Les constructeurs français s' attendent___***V_6_36__1** *à une croissance de 2 % à 3 % en 1992*
159 *Le CCFA (Comité des constructeurs français d' automobiles) a publié___***V_6_375__1** *, mardi 7 janvier , des perspectives 1992 très mesurées___***V_32A_84__1** *.*
160 *L' organisme professionnel prévoit___***V_6_359__1** *une croissance de 2 % à 3 % des ventes de voitures particulières en France pour l' ensemble de l' année 1992 .*
161 *Le marché , qui a enregistré___***V_6_166__1** *en 1991 un repli de 12 % , atteindrait___***V_38L1_10__1** *ainsi les 2 100 000 unités .*

TABLE 10.2: Extrait du sous-corpus `general_lemonde` avec les identifiants des verbes analysés

décider (phrase 2) : table 8 (*Ceci décidera de que ce Luc parte*)
imposer (phrase 2) : table 37M1 (*On impose les revenus des citoyens d'une lourde taxe*)
utiliser (phrase 2) : table 11 (*Max utilise (ses talents+Ida) à faire la cuisine*)
venir (phrase 7.1) : table 2 (*Max vient chez Luc déjeuner*)
nourrir (phrase 7.1) : table 37M1 (*Max nourrit Léa de riz*)
lancer (phrase 7.1) : table 9 *Max lance à Ida qu'il va partir*)
régner (phrase 7.1) : table 35R (*Max règne sur la cuisine*)
souffrir (phrase 8) : table 35R (*Cet appareil souffre d'un défaut de fabrication*)
manquer (phrase 9) : table 34L0 (*Le sel manque dans ce plat*)
effectuer (phrase 54) : table 32R2 (*Cette machine effectue les calculs*)
se rendre (phrase 54) : table 2 (*Max s'est rendu à Paris chercher du travail*)
préciser (phrase 55) : table 9 (*Max a précisé à Luc qu'il faudrait travailler*)
rendre (phrase 59) : table 39 (*Ses diplômes rendent Max un candidat valable*)
publier (phrase 159) : table 6 (*Max a publié que le gouvernement était corrompu*)
prévoir (phrase 160) : table 6 (*Max a prévu que Léa arriverait en retard*)
atteindre (phrase 161) : table 38L1 (*Max atteint (le sommet+la bouteille)*)

TABLE 10.3: Assignations correctes des entrées codées dans les tables

Table erronée donnée par FRMG → Table attendue
se dégrader (phrase 1) : table 32PL (*Max a dégradé les couleurs*) → table 32C (*Les enfants ont dégradé le matériel*) : pas de colonne codant l'entrée pronominale
asphyxier (phrase 2) : table 32H (*Ces gaz ont asphyxié le bébé*) → table 4 (*Ce type d'éducation asphyxie les enfants*)
remplacer (phrase 2) : table 32H (*Cet interne remplace le médecin*) → table 38R (*Max a remplacé le déjeuner par une marche*)
ajouter (phrase 55) : table 10 (*Max a ajouté au dossier que l'accord avait été respecté*) → table 9 (*Max a ajouté à Paul qu'il n'y aurait pas de réunion*) : ambigu car le complément qui permettrait de faire la distinction entre la table 9 et 10 est absent
être (phrase 56) : table 33 (*Cette voiture est à Max*) → table 32NM, entrée 61 (*Max est un idiot*)
révéler (phrase 59) : table 32H (*Ce livre a révélé (cette artiste+Max) au public*) → table 9 (*Max a révélé à Luc que Léa était partie*)
se plaindre (phrase 62) : table 13 (*Max plaint Luc de ce qu'il soit si malheureux*) → table 15 (*Max se plaint amèrement à Luc de ce que tout est dérangé*)
s' élever (phrase 79) : table 38LD (*La crue a élevé le niveau (à la cote 23+de cinquante cm)*) → table 33 (*La facture s'élève à 1000 francs*)
charger (phrase 83) : table 37M6, entrée 63 (*Max charge le camion d'oranges*) → table 13, entrée 23 (*Max charge Luc de ce que tout soit en ordre*)
s' attendre (phrase 157) : table 6 (pas pronominal) → table 7 (*Max s'attend à ce qu'Ida parte*)

TABLE 10.4: Assignations incorrectes d'entrées codées dans les tables

Table erronée donnée par FRMG → Table ou autre entrée attendue
se poursuivre (phrase 4) : table 37M1 (*Max poursuit Ida de ses (sarcasmes+assiduités)*) → pas d'entrée pour ce verbe
avoir (phrase 55) : table 38R (*Paul a eu Luc comme secrétaire*) → verbe support du nom *officiel*, non codé dans les tables
présider (phrase 57) : table 32R3, entrée 692 (*Max préside (le banquet+à table)*) → table 32R3, entrée 693 (*Max préside cette association*) : l'entrée n'est pas codée (codage ∼)
parler (phrase 63) : table 5 (*Ceci parle pour Paul*) → table 1 (*Max parle de prendre sa retraite*) : l'entrée n'est pas codée (codage ∼)
être (phrase 59) : table 5 (*Il est dans les intentions de Max que Léa vienne*) → ne correspond à aucune des entrées des tables : c'est une pseudo-clivée inversée
se poursuivre (phrase 81) : table 6 (*Max poursuivit que la situation était grave et qu'il fallait agir*) → pas d'entrée pour ce verbe
enregistrer (phrase 161) : table 6 (*Luc a enregistré que Luc était passé*) → table 32NM (*On a enregistré une hausse*) : l'entrée n'est pas codée (codage ∼)

TABLE 10.5: Assignations incorrectes d'entrées non codées dans les tables

Remarquons que pour les assignations incorrectes d'entrées codées dans les tables, l'erreur peut aussi bien venir de l'analyseur que des tables. Une étude plus poussée permettrait de le définir. En revanche, pour les assignations incorrectes d'entrées non codées dans les tables, cela montre que le codage de certaines entrées a besoin d'être complété.

2 *A quoi servent___***V_18_16__1** *les ressources linguistiques ?*
7 *Les ressources linguistiques sont utilisées___***V_11_225__1** *pour évaluer___***V_6_197__1** *les systèmes de traitement du langage oral ou écrit existants et établir___***V_39_34__1** *des comparatifs entre systèmes .*
8 *Les évaluations sont particulièrement importantes lorsqu' il s' agit de noter___***V_10_147__1** *les évolutions et progrès d' un système .*
9 *Des secteurs comme ceux des télécommunications , de l' information et de la communication , des affaires internationales (et multilingues) , de l' éducation ou de l' apprentissage sont___***V_5_63__1** *en contact direct et permanent avec les langues , pour la traduction , la terminologie , la reconnaissance de texte ou de la parole , l' extraction , la recherche d' information , etc .*
10 *L' utilisation et l' exploitation des ressources linguistiques , intégrées___***V_38LD_201__1** *ou pas dans des produits spécifiques , apportent___***V_10_15__1** *aux organisations et entreprises qui évoluent___***V_35R_197__1** *dans ces secteurs de nombreux avantages et bénéfices .*
11.1 - *Augmenter___***V_36R_16__1** *la rentabilité .*
11.2 - *Améliorer___***V_4_33__1** *les services et l' information .*
41 *ELRA participe___***V_33_107__1** *ainsi pleinement au développement et à la croissance du marché et du domaine de l' Ingénierie Linguistique .*
48 *L' activité d' évaluation chez ELRA et ELDA a débuté___***V_31H_151__1** *en fournissant___***V_33_68__1** *les ressources linguistiques appropriées___***V_36R_11__1** *pour les tests d' évaluation (Amaryllis , Aurora , CLEF) .*
50 *L' évaluation est___***V_33_61__1** *un facteur de premier plan dans le domaine de l' ingénierie linguistique , et ce à différents niveaux .*
51 *L' évaluation permet___***V_9_283__1** *aux équipes de R & D de valider___***V_32RA_459__2** *leurs recherches et d' assurer___***V_32RA_58__1** *ainsi le développement des systèmes , en constante progression .*
52 *L' évaluation permet___***V_9_283__1** *aussi d' identifier___***V_32R3_467__1** *les créneaux vers lesquels il est judicieux de s' orienter___***V_35R_290__1** *pour développer___***V_32R3_290__1** *de nouvelles technologies , aboutissant___***V_35LD_3__1** *à leur mise sur le marché .*
54 *Les campagnes d' évaluation contribuent___***V_7_37__1** *également à la production de données de test ou d' entraînement de très bonne qualité , de méthodologies , métriques et protocoles , qui peuvent___***V_1_88__1** *ensemble être mis___***V_38LR_158__1** *à disposition sous la forme de ' lots pour l' évaluation' , et distribués___***V_32R3_304__1** *comme le sont___***V_35R_195__1** *les ressources linguistiques habituelles .*
56 *L' activité d' évaluation peut___***V_1_92__1** *être menée___***V_3_29__1** *à différents niveaux :*
56.1 - *Evaluation des technologies (pour trouver___***V_11_122__1** *les solutions aux problèmes clairement définis)*
57 *Les projets d' évaluation auxquels ELRA/ELDA participe___***V_33_106__1** *à l' heure actuelle sont___***V_31H_225__1** *: - Technolangue/EVALDA*

Table 10.6: Extrait du sous-corpus `general_elda` avec les identifiants des verbes analysés

utiliser (phrase 7) : table 11 (*Max utilise (ses talents+Ida) à faire la cuisine*)
évaluer (phrase 7) : table 6 (*Max évalue si l'opération est rentable ou non*)
intégrer (phrase 10) : table 38LD (*Jo a intégré cet appareil dans son système*)
évoluer (phrase 10) : table 35R (*La maladie évolue (E+vers) un mieux*)
permettre (phrase 51) : table 9, entrée 283 (*Ceci permet à Luc qu'il ait ce poste*)
valider (phrase 51) : table 32RA (*Max a validé le document (E+d'un tampon)*)
assurer (phrase 51) : table 32RA (*Max assure sa prise*)
permettre (phrase 52) : table 9, entrée 283 (*Ceci permet à Luc qu'il ait ce poste*)
orienter (phrase 52) : table 35R (*Max s'oriente (sur+d'après) l'étoile polaire*)
contribuer (phrase 54) : table 7 (*Max contribue à ce que tout soit en ordre*)
pouvoir (phrase 56) : table 1 entrée 88 (*Max pourrait réussir, qui sait ?*)

Table 10.7: Assignations correctes d'entrées codées dans les tables

Table erronée donnée par FRMG → Table attendue
servir (phrase 2) : table 18 (*Qu'Ida soit partie sert de prétexte à Max pour la congédier*) → table 16 (*Ce torchon sert à Max à nettoyer les vitres*) ou table 14 (*Cette histoire a servi pour Max à ce qu'il soit élu*)
établir (phrase 7) : table 39 (*Max a établi son fils (E+comme) notaire*) → table 32A (*Max a établi une liste de suspects*)
noter (phrase 8) : table 10 (*Max note sur son agenda (qu'il aille+d'aller) chez le coiffeur*) → table 6 (*Max note (que Léa est absente+son impatience)*)
améliorer (phrase 11.2) : table 4 (*Qu'il ait travaillé améliore Max*) → table 32RA (*Max améliore le potage d'un peu de poivre*)
participer (phrase 41) : table 33, entrée 107 (*Je participe à votre (douleur+joie)*) → table 33, entrée 108 (*Max participe (au jeu+aux délibérations)*) : c'est uniquement la construction de base *N0 V à N1* qui permet de les reconnaître, sans les distinguer
débuter (phrase 48) : table 31H (*Max débute dans (la vie+le monde)*) → table 38R (*Max débute son repas par un oeuf*)
fournir (phrase 48) : table 33 (*Sa famille fournit aux dépenses de Bob*) → table 36DT (*Max fournit du vin à Luc*)
être (phrase 50) : table 33 (*Cette voiture est à Max*) → table 32NM, entrée 62 (*Max est un idiot*)
identifier (phrase 52) : table 32R3 (*Max a identifié le corps*) → table 6 (*Max identifiera si le tableau est vrai ou non*)
développer (phrase 52) : table 32R3, entrée 209 (*Max a développé ce produit*) → table 32R2 (*Max développe (ses ventes+son usine)*)
aboutir (phrase 52) : table 35LD (*Max aboutit au village*) → table 14 (*Tes ordres ont abouti pour Luc à ce qu'il se révolte*)
pouvoir (phrase 54) : table 1, entrée 88 (*Max pourrait réussir, qui sait ?*) → table 1, entrée 87 (*Max peut passer entre les barreaux*)
distribuer (phrase 54) : table 32R3 (*Le metteur en scène a distribué (la pièce+le film)*) → table 36DT (*Max distribue de l'argent à ses amis*)
participer (phrase 57) : table 33, entrée 106 (*Ces colonnes participent à la beauté de l'édifice*) → table 33, entrée 108 (*Max participe (au jeu+aux délibérations)*)
être (phrase 57) : table 31H (*Max en est*) → table 32NM, entrée 62 (*Max est un idiot*)

Table 10.8: Assignations incorrectes d'entrées codées dans les tables

Table erronée donnée par FRMG → Table ou autre entrée attendue
être (phrase 9) : table 5 (*Il est dans les intentions de Max que Léa vienne*) → verbe support *être en*, non codé dans les tables
apporter (phrase 10) : table 10 (*Cette remarque a apporté dans la discussion que tout devait être revu*) → table 36DT (*Paul apporte des fleurs à Marie*) : l'entrée n'est pas codée (codage ∼)
augmenter (phrase 11.1) : table 36R (*Max a augmenté le nombre des lits à six*) → table 37M4 (*Max a augmenté son rapport de trois pages*) : l'entrée n'est pas codée (codage ∼)
mettre (phrase 54) : table 38LR (*Max met cette ville dans le Sud*) → expression figée *mettre à disposition*, non codée dans les tables
être (phrase 54) : table 35R, entrée 195 (*Max en est de cent francs dans cette histoire*) → *être* est auxiliaire de *distribuer* : il devrait être reconnu par le Le*fff* mais il est difficile à reconnaître dans cette phrase
mener (phrase 56) : table 3 (*Max mène Ida se faire coiffer*) → verbe support du nom *activité*, non codé dans les tables
trouver (phrase 56.1) : table 11, entrée 122 (*Max a trouvé du plaisir à faire cela*) → table 32A (*Max a trouvé un (nouveau médicament+algorithme)*) : l'entrée n'est pas codée (codage ∼)

TABLE 10.9: Assignations incorrectes d'entrées non codées dans les tables

Nombre d'entrées dans les tables	1	2	3	4	5	6 à 27	Total
Taux de succès	100%	50%	67%	50%	10%	32%	41%
Taux de succès parmi les entrées codées dans les tables	100%	50%	75%	50%	17%	50%	52%

TABLE 10.10: Taux de succès dans les deux sous-corpus en fonction du nombre d'entrées de chaque verbe

Pour finir, nous avons testé quelques noms prédicatifs sans déterminant, pour vérifier s'ils étaient bien reconnus par FRMG. Les noms prédicatifs sans déterminant ne représentent qu'une infime partie des entrées du Lexique-Grammaire ; en effet, uniquement 241 entrées acceptent la propriété Det = : <E> ou ont la valeur <E> dans les colonnes autre Det ou Det figé. Nous avons testé quelques phrases non tirées de corpus [9] dont les résultats sont à la Tab. 10.11.

Mode désambiguïsé	Autres possibilités en mode non désambiguïsé
Assignation correctes d'entrées : *Max a soin*___**N_an01_120__2** *de ses affaires* *Max a hâte*___**N_ape3_86__2** *de partir* *Max fait abus*___**N_f8_1__1** *de cette règle auprès de Marie* *Max fait honneur*___**N_fnan_116__1** *à Marie* *Max fait allusion*___**N _fnpn_3__1** *à cette série* *Max donne abri*___**N_dr1_1__1** *à Marie*	
Assignation d'entrées codées dans plusieurs tables : *Max a accès*___**N_ape2_1__2** *à la sortie* *Max a droit*___**N_ana_77__2** *à des sous*	 *Max a accès*___**N_ana_77__2** *à la sortie* *Max a droit*___**N_ape2_20__2** *à des sous* *Max a droit*___**N_ape2_21__2** *à des sous* *Max a droit*___**N_ape2_22__2** *à des sous*
Assignation d'entrées verbales de la table 32NM : *La bombe a fait*___**V_32NM_65__1** *explosion* *Max a*___**V_32NM_12__1** *mal à la tête*	 *La bombe a fait explosion*___**N_f1b_19__1** *Max a mal*___**N_anml_198__1** *à la tête*
Assignation d'entrées du Le*fff* : *Max fait honte à Marie* *La porte fait obstacle au passage*	 *Max fait honte*___**N_f41_8__1** *à Marie* *La porte fait obstacle*___**N_fs2_71__1** *au passage*

TABLE 10.11: Phrases avec les identifiants des noms (ou verbes) analysés

Nous constatons que, dans l'ensemble, les assignations d'entrées sont correctes. Certaines entrées sont codées dans plusieurs tables (ici APE2 et ANA) et donc sont reconnues par une seule arbitrairement lors de la désambiguïsation. D'autres ne sont pas choisies lors de la désambiguïsation à cause des verbes de la table 32NM. Enfin, certaines sont directement reconnues par le Le*fff*, puisque nous n'avons pas exclu les quelques noms prédicatifs que contient le Le*fff*.

10.5 Conclusion

Nous avons développé une méthodologie et un outil permettant de convertir la version textuelle des tables du Lexique-Grammaire en un lexique pour le TAL utilisant le forma-

9. En faisant une recherche dans les deux sous-corpus `general_lemonde` et `general_elda`, la seule phrase contenant un des noms prédicatifs cités à la Tab. 10.11 est :
2222 *Aucun des commentaires publiés n' a fait allusion*___**N _fnpn_3__1** *à cet aspect des relations entre Paris et Pékin .*
Cela explique pourquoi dans les 35 phrases analysées ci-dessus aucun nom prédicatif sans déterminant n'a été trouvé.

lisme lexical du Le*fff*, ce qui permet de l'utiliser à peu près immédiatement par l'analyseur syntaxique FRMG. La pertinence du lexique obtenu a été validée par son utilisation pour l'analyse syntaxique automatique du corpus d'évaluation de la campagne Passage. En effet, pour un lexique de bonne qualité comme les tables du Lexique-Grammaire, les résultats obtenus sont bons.

Ce travail nous a permis d'identifier un certain nombre de problèmes dans les données de départ (tables et table des classes), mais il nous a également conduit à un certain nombre de simplifications et d'approximations dans le processus de conversion. Il subsiste d'importantes possibilités d'améliorations, qui pourraient permettre de construire un lexique syntaxique pour le TAL à partir des tables du Lexique-Grammaire, avec en perspective l'amélioration de la qualité des outils et des ressources existants. Ceci par fusion avec d'autres ressources lexicales, par exemple, ou par intégration dans un analyseur syntaxique opérationnel.

11 Conclusion et perspectives

Nous dressons un bilan du travail effectué sur les tables du Lexique-Grammaire du français afin de les rendre exploitables dans des analyseurs syntaxiques dans la section 11.1. Puis, nous évoquons en 11.2.1 l'extension de ce travail à d'autres langues, telles que le grec, afin de construire un lexique syntaxique à partir des tables du Lexique-Grammaire du grec. Enfin, nous terminons par quelques pistes afin d'améliorer le lexique *LGLex*-Le*fff* du français et son intégration dans FRMG en 11.2.2.

11.1 Conclusion

Les tables du Lexique-Grammaire du français constituent une grande base de données lexicales, syntaxiques et sémantiques. Diverses incohérences et carences empêchaient leur exploitation automatique directe dans des outils de TAL :

- des intitulés de colonnes qui différaient d'une table à l'autre, bien que dénotant la même propriété linguistique ;
- des intitulés de colonnes qui ne dénotaient pas la même propriété d'une table à l'autre ;
- des intitulés de colonnes qui laissaient implicites des informations, pourtant nécessaires à leur exploitation automatique ;
- des informations importantes qui n'étaient pas représentées, car elles étaient considérées comme implicites pour une table donnée.

Pour que les tables forment un tout cohérent, nous avons réalisé un ensemble de modifications :

- nouvelles tables numérisées et corrigées pour les noms prédicatifs et les expressions figées ;
- renommage dans toutes les tables d'un grand nombre de propriétés ;

- création de la table des classes pour chaque catégorie regroupant toutes les propriétés ;
- définition des propriétés définitoires vraies pour toutes les entrées d'une table, ainsi que leur codage dans la table des classes concernée ;
- codage de propriétés dans les tables et dans la table des classes ;
- ajout d'entrées dans les tables de verbes et déplacement de certaines entrées ;
- ajout d'exemples pour les tables de noms ;
- documentation de toutes les propriétés des tables de verbes ;
- définition formelle de toutes les tables de verbes ;
- arbre de classement permettant de déterminer à quelle classe appartient une entrée verbale donnée.

De plus, nous avons décrit tous les changements que nous avons réalisé pour définir les constructions de base pour chaque classe dans toutes les catégories (verbes distributionnels, noms prédicatifs, expressions figées et adverbes).

Nous avons converti les tables au format *LGLex* pour l'ensemble des catégories grâce à l'outil *LGExtract*. Cet outil, associé à un script et à la table des classes, permet de créer un lexique au format XML ou texte. Ce script décrit les opérations à effectuer pour chaque propriété, ce qui permet de définir une seule fois chaque intitulé. Ce lexique manipule les mêmes concepts que ceux des tables, il est donc utilisable par toute personne voulant utiliser les tables sans devoir lire des tables Excel, qui ne servent qu'à la lisibilité lors de leur visualisation.

Enfin, nous avons converti les tables des verbes et des noms prédicatifs au format Alexina, le format du lexique Le*fff*, afin de pouvoir les utiliser directement dans l'analyseur syntaxique FRMG. Pour cela, nous avons développé une méthodologie et un outil permettant de convertir la version textuelle des tables du Lexique-Grammaire en un lexique pour le TAL utilisant le formalisme lexical du Le*fff*, ce qui le rend utilisable par l'analyseur syntaxique FRMG.

De plus, ce travail de conversion nous a permis d'identifier un certain nombre de problèmes dans les données de départ (tables et table des classes), que nous avons pu corriger.

La pertinence du lexique obtenu a été validée par son utilisation pour l'analyse syntaxique automatique du corpus de référence de la première campagne Passage. En effet, nous avons évalué l'analyseur FRMG avec le lexique que nous avons créé (*LGLex*-Le*fff*), tout en conservant le Le*fff*, hormis les entrées verbales. Nous en avons conclu que les tables de verbes permettent d'obtenir d'aussi bons résultats que l'état de l'art.

11.2 Perspectives

11.2.1 Application aux tables du grec

En excluant le français, le grec est la langue pour laquelle il existe le plus de tables du Lexique-Grammaire. C'est pourquoi nous envisageons leur conversion en un lexique

syntaxique (Ioannidou et Kyriacopoulou, 2010).

Les tables du Lexique-Grammaire du grec sont en cours d'élaboration et constituent la principale source d'informations lexicales, syntaxiques et sémantiques de la langue. Il existe aujourd'hui 17 tables de verbes simples (Kyriacopoulou, 2010), 7 de noms prédicatifs (Sfetsiou, 2007) et 16 tables d'adverbes (semi-)figés (Voyatzi, 2006; Voyatzi et Kakoyianni-Doa, 2010).

Les tables verbales concernent les verbes transitifs non locatifs (tables 32GA, 32GD, 32GC, 32GCL, 32GCV, 32GH, 32GNM, 32GPL, 32GRA), les tables à complétives (4G, 6G et 9G) ainsi que les verbes transitifs locatifs (tables 38GL, 38GLS, 38GLD, 38GLH, 38GLR). Les tables des noms prédicatifs concernent les noms à verbe support *κάνω*, *faire* en français (tables FN1, FN2, FNPN, FNAN-hum, FNANhum, FNANlieu, FND-NAN). Les tables d'adverbes (semi-)figés sont parfaitement analogues aux 16 tables des adverbes figés du français (Gross, 1990). Elles portent les mêmes noms que celles du français, précédés de la lettre G pour «grec» (tables GPADV, GPC, GPAC, GPCA, GP-CONJ, GPCPC, GPCDC, GPCONJ, GPCDN, GPCPN, GPV, GPF, GPECO, GPVCO, GPPCO, GPJC).

Les tables verbales comptent 3 969 entrées, les tables de noms prédicatifs 900 entrées, et les tables d'adverbes (semi-)figés 5 080 entrées.

La démarche générale est la suivante :
- collecter toutes les tables du grec ;
- créer une table des classes par catégorie grammaticale, afin de rendre cohérent l'ensemble des tables et d'y faire figurer les propriétés définitoires ;
- créer un script par catégorie grammaticale en interprétant chaque intitulé, afin de convertir les tables au format *LGLex* ;
- modifier *lglex2ilex.pl* en tenant compte des différences dans la langue grecque, afin de convertir les tables au format Alexina, le format du lexique Le*fff*.

Enfin, si l'on souhaite utiliser ce lexique syntaxique dans un analyseur syntaxique, il faut adapter la méta-grammaire du français à celle du grec, comme cela a été fait pour l'espagnol (Fernández González, 2010).

Nous avons déjà participé à l'harmonisation des tables du grec avec la collaboration de K. Ioannidou, dans le but de constituer la table des classes des verbes grecs, incluant toutes les propriétés définitoires [1]. Lors de la création de la table des classes, nous avons

1. Nous avons défini les propriétés définitoires vraies pour l'ensemble des entrées de chaque tables, incluant la construction de base, comme suit :
32GA : N0 V N1, avec N1 apparition
32GD : N0 V N1, avec N1 disparition
32GC : N0 V N1, avec N1 =: Nconc
32GCL : N0 V N1, avec N1 =: Npc
32GCV : N0 V N1, avec N0 Vsup N1 Prep V-n
32GH : N0 V N1, avec N1 =: Nhum
32GNM : N0 V N1, avec N1 =: N-hum
32GPL : N0 V N1, avec N1 =: Npl obl
32GRA : N0 V N1, avec N0 Vsup N1 V-adjaccusatif *[équivalent à* V-adj*]*
4G : N0 V N1, avec une complétive en N0

rencontré divers types d'ambiguïtés formelles.

À part des problèmes typographiques divers (présence ou non d'espaces, alternance de lettres majuscules et minuscules, ou de caractères grecs et latins), certains intitulés exprimant la même notion linguistique sont notés avec des symboles différents (par exemple, N1 = : Nplur obl et N1 = pluriel) ou avec des conventions différentes [2] (Ioannidou et Kyriacopoulou, 2010). Nous avons choisi de garder au maximum les notations utilisées dans les tables du Lexique-Grammaire du français. La table des classes contenait au départ 280 propriétés, incluant les différentes notations. À partir de la table des classes, nous avons repéré les erreurs de notation et nous les avons corrigées directement dans les tables. La nouvelle table des classes générée contient 202 propriétés.

Cette étape, achevée pour les verbes, est en cours pour les noms prédicatifs. Le script de conversion des verbes est également en cours de réalisation, ce qui nous permettra de générer une première version du lexique *LGLex* des verbes grecs (Ioannidou et Tolone, 2011).

L'objectif est d'homogénéiser, corriger et compléter les données pour l'ensemble des tables du Lexique-Grammaire du grec. Une fois ces tables syntaxiques corrigées, elles seront converties au format *LGLex*, afin d'être exploitables dans des analyseurs syntaxiques.

6G : N0 V N1, avec une complétive en N1
9G : N0 V N1 σε *[se=à]* N2, avec une complétive en N1
38GL : N0 V N1 Loc N2 source Loc N3 destination
38GLS : N0 V N1 Loc N2 source
38GLD : N0 V N1 Loc N2 destination
38GLH : N0 V N1 Loc N2 destination, avec N1 =: Nhum
38GLR : N0 V N1 Loc N2
FN1 : N0 κάνω *[káno=faire]* N
FN2 : N0 κάνω *[káno=faire]* N Pνα *[Pna=P avec conjonction na, suivie de phrases au subjonctif]*
FNPN : N0 κάνω *[káno=faire]* N με *[me=avec]* N1
FNAN-hum : N0 κάνω *[káno=faire]* N σε *[se=à]* N1, avec N1 = : N-hum
FNANhum : N0 κάνω *[káno=faire]* N σε *[se=à]* N1, avec N1 = : Nhum
FNANlieu : N0 κάνω *[káno=faire]* N σε *[se=à]* N1, avec N1 locatif
FNDNAN : N0 κάνω *[káno=faire]* N N1genitif *[équivalent à* de N1*]* σε *[se=à]* N2
Nous n'indiquons pas les structures de base des adverbes (semi-)figés, puisque ce sont les mêmes que pour le français.

2. Nous avons trois manières d'exprimer ce qui est optionnel en utilisant des parenthèses. D'une part, on peut mettre uniquement la partie optionnelle entre parenthèses et dans ce cas les parenthèses sont interprétés comme permettant d'inclure l'optionnel (N0 V N1 (Loc N2 source) Loc N3 destination). D'autre part, on peut mettre entre parenthèses la structure optionnelle suivie du symbole +, suivie ou précédé du symbole vide E, représentant la chaîne vide (N0 V N1 (Loc N2 source+E) Loc N3 destination et N0 V N1 (E+Loc N2 source) Loc N3 destination). Dans ce deuxième cas, les parenthèses sont interprétées comme incluant plusieurs sous-structures au choix, le choix étant exprimé par le symbole +. C'est la troisième notation qui a été choisie (c'est-à-dire le symbole + précédé du symbole E), afin de rester conforme aux notations du français.

11.2.2 Améliorer le lexique *LGLex*-Le*fff* du français et son intégration dans FRMG

Comme nous l'avons mentionné tout au long de cette thèse, le travail entrepris pour le français doit être poursuivi :

- des entrées sont à ajouter ou à coder dans les tables existantes (cf. 5.5.2) ; citons par exemple le codage des paraphrases des adverbes en *-ment* (notamment, le codage des propriétés lexicales associées) présentées en 6.5.1, ainsi que l'intégration des variantes des entrées adverbiales dans le lexique *LGLex*, comme montré dans la section 7.3 (Tolone et Voyatzi, 2011) ;
- de nouvelles tables sont à créer, ou simplement à numériser et corriger pour certaines encore restantes (cf. 5.2.4 et 5.5.2), comme les tables de L. Pivaut (Tolone, 2011) ;
- des propriétés sont à coder, aussi bien dans la table des classes que dans les tables elles-mêmes (cf. 5.5.2) ; de plus, certaines propriétés sont à ajouter, comme par exemple certaines constructions, notamment pronominales ou ayant la formation en général régulière (comme la négation ou les propositions relatives) ;
- les processus de conversion sont à améliorer, pour corriger les erreurs et tenir compte de plus d'informations ;
- une évaluation plus fine est à réaliser afin de détecter précisément d'où viennent les erreurs d'analyse, permettant ainsi d'améliorer les tables et/ou l'analyseur FRMG (Tolone *et al.*, 2011).

Tout d'abord, le travail de conversion nous a permis d'identifier un certain nombre de problèmes dans les données de départ (tables et table des classes), mais il nous a également conduit à un certain nombre de simplifications et d'approximations dans le processus de conversion.

Il subsiste en effet d'importantes possibilités d'améliorations :

- identifier et corriger les erreurs qui restent certainement encore à chacune des étapes du processus ;
- tenir compte de certaines constructions qui sont ignorées ;
- mieux gérer les cas difficiles comme les entrées ayant une construction de base avec une ou deux constructions définitoires, telles que les tables de symétriques ou de converses ;
- récupérer la classe de fréquence dans le DELA, afin de la traduire en un poids de la façon suivante : les entrées **z1** reçoivent un poids standard de *100*, les entrées **z2** reçoivent un poids de *70*, et les entrées **z3** reçoivent un poids de *50*.
- intégrer les déterminants (et la présence de modifieurs) qui sont codées dans les tables de noms prédicatifs, qui sont jusqu'à présent ignorés ;
- ajouter les informations sur la réduction du groupe nominal des noms prédicatifs, afin de clarifier l'attachement dans les groupes nominaux ;
- finir la conversion des tables d'expressions figées qui a été amorcée (cf. 8.1.6), ce qui demande une modification complète des entrées dans les tables afin de déterminer leurs différents constituants et leur catégorie ;
- convertir également les tables d'adverbes, en considérant toutes les variantes

comme des entrées supplémentaires (cf. section 7.3).

Cette thèse a permis de construire un lexique syntaxique pour le TAL à partir des tables du Lexique-Grammaire, avec en perspective l'amélioration de la qualité des outils et des ressources existants.

En effet, l'intégration dans l'analyseur FRMG doit être poursuivie :
- coupler les deux variantes de l'analyseur (FRMG_{LGLex}et $\text{FRMG}_{\text{Le}fff}$) pour garder un maximum d'analyses complètes, bien meilleures que les analyses partielles (cf. section 10.2) ;
- détecter des erreurs, notamment au moyen de techniques automatiques, telles que la fouille d'erreurs dans les sorties d'analyseurs syntaxiques (Sagot et de La Clergerie, 2008), en poursuivant le travail présenté dans la section 10.3 (Tolone *et al.*, 2011) ;
- prendre en compte les noms prédicatifs avec déterminants, ce qui permettra une meilleure évaluation du lexique *LGLex*-Le*fff* puisque les noms prédicatifs sans déterminant, les moins nombreux dans les corpus, sont les seuls qui ont pu être évalués (cf. section 10.4). Du point de vue de l'analyse, autoriser des déterminants sur les noms prédicatifs fait exploser le nombre d'analyses ; il est possible qu'une bonne approche soit une réécriture en sortie d'analyse syntaxique (dans la phase de désambiguïsation par exemple) ;
- prendre en compte des informations sémantiques (macros du type $@fNhum$ ou $@fNhum$, où f est la fonction syntaxique) jusqu'à présent ignorées, ou d'autres informations, comme par exemple la macro *@neg*, qui indique le caractère obligatoirement négatif du verbe (cf. 8.1.4) ; une possibilité est de factoriser les entrées et de choisir une entrée particulière très tardivement ;
- ajouter des constructions non gérées ;
- prendre en compte les informations syntaxiques des adverbes, ignorées jusqu'à présent ; les adverbes peuvent en effet être insérés à n'importe quel endroit de la phrase, sans aucune contrainte.

Enfin, ce travail doit aussi renforcer la fusion de diverses ressources lexicales en une seule ressource de très grande qualité, même si nous devons garder à l'esprit que les choix linguistiques derrière chaque ressource ont un impact : ainsi, le Le*fff* fournit des entrées verbales plutôt factorisées (peu de distinctions sémantiques, cadres de sous-catégorisation factorisés) alors que *LGLex* liste de nombreuses entrées par verbes correspondant à divers sens et associées à des cadres de sous-catégorisation plus simples mais se chevauchant.

Nous pouvons envisager d'améliorer *LGLex*-Le*fff* en le fusionnant avec, par exemple, le Le*fff* et/ou DICOVALENCE, en employant une méthode similaire à celle de Danlos et Sagot (2008) :
- conversion des ressources de départ au format Le*fff* ;
- regroupement des entrées des différentes ressources ;
- fusion des regroupements d'entrées.

Pour un lemme donné, chaque ressource est susceptible de comporter plusieurs entrées. Il faut donc déterminer le nombre d'entrées que l'on souhaite obtenir au final, afin de

construire ces ensembles d'entrées provenant des différentes ressources, appelés regroupements.

Les entrées qui forment un regroupement doivent ensuite être fusionnées :

- l'ensemble des fonctions syntaxiques est construit comme l'union des ensembles des fonctions syntaxiques présentes dans les entrées sources ;
- pour chaque fonction, l'ensemble des réalisations est également obtenu par union, en indiquant quelle source propose quelle réalisation ;
- une fonction syntaxique n'est à réalisation obligatoire que si c'est le cas dans toutes les ressources sources ;
- l'ensemble des redistributions possibles est construit comme l'union des ensembles de redistributions possibles présentes dans les entrées sources.

Ceci doit s'accompagner d'une validation manuelle, notamment des entrées qui se sont multipliées par rapport au nombre initial d'entrées du lemme concerné dans les tables du Lexique-Grammaire (étant considérée comme la ressource ayant le plus d'entrées par lemme).

L'utilisation d'une ressource lexicale la plus riche possible reste un moyen efficace pour améliorer la qualité d'un analyseur syntaxique (Sagot et de La Clergerie, 2008). Ceci constitue une des motivations pour poursuivre la construction et l'amélioration des tables du Lexique-Grammaire du français et d'autres langues.

Bibliographie

Abeillé, A. (2002). *Une grammaire électronique du français.* CNRS Editions, Paris, France. 23, 52

Abeillé, A. (2003). *Treebanks : Building and Using Parsed Corpora*, volume 20 de *Text, Speech and Language Technology.* Kluwer Academic Publishers, Dordrecht, Pays-Bas. 30

Aït-Mokhtar, S., Chanod, J.-P. et Roux, C. (2002). Robustness beyond shallowness : incremental deep parsing. *Natural Language Engineering*, 8(3):121–144. 22, 26

Alcouffe, P., Revellin-Falcoz, B. et Zaysser, L. (1993). Azote : des tables du LADL au format Genelex. *In Actes du colloque Informatiques et Langues Naturelles (ILN'93)*, IRIN, Université de Nantes, France. 70

Alonso, M. et Díaz, V. (2003). Variants of mixed parsing of TAG and TIG. *Traitement Automatique des Langues (T.A.L.)*, 44(3):41–165. 56

Alonso, M. A., Cabrero, D., Vilares, M. et de La Clergerie, E. (1999). Tabular algorithms for TAG parsing. *In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 150–157. 56

Baker, C., Fillmore, C. et Lowe, J. (1998). The Berkeley FrameNet project. *In Proceedings of ACL-CoLing'98*, Montreal, Canada. 14, 31, 70

Béchet, N. (2009). *Extraction et regroupement de descripteurs morpho-syntaxiques pour des processus de Fouille de Textes.* Thèse de doctorat, Université Montpellier 2, Montpellier, France. 22, 25

Blache, P. et Rauzy, S. (2006). Mécanismes de contrôle pour l'analyse en grammaires de propriétés. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'06)*, pages 415–424, Louvain, Belgique. 25

Blanc, O. (2006). *Algorithmes d'analyse syntaxique par grammaires lexicalisées : optimisation et traitement de l'ambigüité.* Thèse de doctorat, IGM, Université Paris-Est Marne-la-Vallée, France. 156

Blanche-Benveniste, C., Delofeu, J., Stefanini, J. et van den Eynde, K. (1987). *Pronom et syntaxe : L'approche pronominale et son application au français*. SELAF, Paris, France. 27

Boons, J.-P., Guillet, A. et Leclère, C. (1976a). La structure des phrases simples en français : Classes de constructions transitives. Rapport technique, LADL, CNRS, Paris 7. 39, 67, 82, 100, 101, 102, 103, 104, 105, 107, 110

Boons, J.-P., Guillet, A. et Leclère, C. (1976b). *La structure des phrases simples en français : Constructions intransitives*. Droz, Genève, Suisse. 39, 67, 76, 77, 92, 100, 101, 104, 108, 110, 135, 200, 277, 293, 294, 299, 300

Borillo, A. (1971). Remarques sur les verbes symétriques français. *Langue française*, 11(1):17–31. 67, 109

Boullier, P. et Sagot, B. (2005). Analyse syntaxique profonde à grande échelle : SxLFG. *Traitement Automatique des Langues (T.A.L.)*, 46(2):65–89. 22, 23

Bourigault, D. et Frérot, C. (2005). Acquisition et évaluation sur corpus de propriétés de sous-catégorisation syntaxique. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'05)*, Dourdan, France. 26

Briscoe, T. et Carroll, J. (1993). Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–59. 26

Cabrera, I. (2008). *Alpage Linguistic Processing Chain for French*. http ://alpage.inria.fr/docs/alpchain-doc.pdf. 45, 52

Candito, M.-H. (1999). *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées*. Thèse de doctorat, Université Paris 7, France. 53

Carroll, J. et Fang, A. C. (2004). The automatic acquisition of verb subcategorisations and their impact on the performance of an HPSG parser. *In* Su, K.-Y., ichi Tsujii, J., Lee, J.-H. et Kwong, O. Y., éditeurs : *IJCNLP*, volume 3248, pages 646–654. Springer. 26

Cervelle, J., Forax, R. et Roussel, G. (2006). Tatoo : an innovative parser generator. *In Proceedings of the conference on Principles and Practices of Programming in Java (PPPJ'06)*, Mannheim, Allemagne. 158

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, États-Unis. 38

Clément, L. (2001). XLFG : a parser to learn the LFG framework. *In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL'01)*, Pittsburgh, États-Unis. 24

Clément, L., Sagot, B. et Bernard, L. (2004). Morphology based automatic acquisition of large-coverage lexica. *In Proceedings of the 4th Language Resource and Evaluation Conference (LREC'04)*, pages 1841–1844, Lisbonne, Portugal. 46

Collins, M. (2003). Head-driven statistical models for Natural Language Parsing. *Computational Linguistics*, 29(4):589–637. 20

Constant, M. et Tolone, E. (2010). A generic tool to generate a lexicon for NLP from Lexicon-Grammar tables. *In* Gioia, M. D., éditeur : *Actes du 27e Colloque international sur le lexique et la grammaire (L'Aquila, 10-13 septembre 2008), Seconde partie*, volume 1 de *Lingue d'Europa e del Mediterraneo, Grammatica comparata*, pages 79–193. Aracne, Rome, Italie. ISBN 978-88-548-3166-7. 83, 94, 155, 156, 161, 243

Crabbé, B. (2005). *Représentation informatique de grammaires d'arbres fortement lexicalisées : le cas de la grammaire d'arbres adjoints*. Thèse de doctorat, Université Nancy 2, France. 24

Crabbé, B. et Candito, M. (2008). Expériences d'analyse syntaxique statistique du français. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08)*, pages 45–54, Avignon, France. 20

Danlos, L. (1980). *Représentation d'informations linguistiques : les constructions N être Prép X*. Thèse de doctorat, Université Paris 7, France. 67, 131, 136, 137

Danlos, L. (1985). *La génération automatique de textes en langue naturelle*. Masson, Paris, France. 26

Danlos, L. (1988). Les phrases à verbe support *être prep*. *Langages*, 23(90):23–37. 123, 137, 138

Danlos, L. et Sagot, B. (2008). Constructions pronominales dans Dicovalence et le Lexique-Grammaire : Intégration dans le Le*fff*. *In* Constant, M., Gioia, M. D., Nakamura, T. et Vecchiato, S., éditeurs : *Actes du 27ème Colloque Lexique et Grammaire (LGC'08)*, L'Aquila, Italie. 47, 156, 181, 226

de Chalendar, G. et Nouvel, D. (2009). Modular resource development and diagnostic evaluation framework for fast NLP system improvement. *In Proceedings of the Workshop SETQANLP on North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT'09)*, Boulder, États-Unis. 22, 24

de La Clergerie, E. (2002). Construire des analyseurs avec DyALog. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'02)*, Nancy, France. 23, 56

de La Clergerie, E. (2005a). DyALog : a tabular logic programming based environment for NLP. *In Proceedings of 2nd International Workshop on Constraint Solving and Language Processing (CSLP'05)*, Barcelone, Espagne. 23, 52, 56, 58

DE LA CLERGERIE, E. (2005b). From metagrammars to factorized TAG/TIG parsers. *In Proceedings of IWPT'05 (poster)*, pages 190–191, Vancouver, Canada. 22, 23, 52, 54

DE LA CLERGERIE, E., SAGOT, B., NICOLAS, L. et GUÉNOT, M.-L. (2009). FRMG : évolutions d'un analyseur syntaxique TAG du français. Communication à la journée ATALA : Quels analyseurs syntaxiques pour le français ? (http ://www.atala.org/Quels-analyseurs-syntaxiques-pour). 56, 57, 60, 61

DE NÉGRONI-PEYRE, D. (1978). Nominalisations par être en et réflexivation (admiration, opposition, révolte et rage). *Lingvisticæ Investigationes*, 2(1):127–164. 67, 125

DENDIEN, J. et PIERREL, J.-M. (2003). Le trésor de la langue française informatisé : un exemple d'informatisation d'un dictionnaire de langue de référence. *Traitement Automatique des Langues (T.A.L.)*, 44(2):11–37. 29, 155

DUBOIS, J. et DUBOIS-CHARLIER, F. (1997). *Les verbes français*. Larousse, Paris, France. 28, 29, 46

DUCHIER, D., LE ROUX, J. et PARMENTIER, Y. (2005). XMG : Un compilateur de métagrammaires extensible. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'05)*, Dourdan, France. 24

FERNÁNDEZ GONZÁLEZ, D. (2010). Cadena de procesamiento lingüístico para el español. Mémoire de D.E.A., Université de Vigo, Espagne. 46, 52, 223

FRANCOPOULO, G. (2008). Tagparser : well on the way to iso-tc37 conformance. *In Proceedings of the 1st First International Conference on Global Interoperability for Language Resources (ICGL'08)*, Hong-Kong, Chine. 22, 24

FRANCOPOULO, G., MONTE, G., CALZOLARI, N., MONACHINI, M., BEL, N., PET, M. et SORIA, C. (2006). Lexical Markup Framework (LMF). *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, Gênes, Italie. 70, 156

GARDENT, C., GUILLAUME, B., PERRIER, G. et FALK, I. (2005). Extracting subcategorisation information from Maurice Gross' grammar lexicon. *Archives of Control Sciences*, 15(3):253–264. In Memoriam Maurice Gross. Special issue on Human Language Technologies as a challenge for Computer Science and Linguistics. Part I. (2nd Language and Technology Conference). 69, 78, 156

GARDENT, C., GUILLAUME, B., PERRIER, G. et FALK, I. (2006). Extraction d'information de sous-catégorisation à partir des tables du LADL. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'06)*, Louvain, Belgique. 69, 86, 90, 156, 157, 172

GIRY-SCHNEIDER, J. (1978). *Les nominalisations en français : L'opérateur faire dans le lexique*. Droz, Genève, Suisse. 67, 118, 119, 122

Giry-Schneider, J. (1987). *Les prédicats nominaux en français : Les phrases simples à verbe support*. Droz, Genève, Suisse. 67, 120, 121, 122, 131, 135

Giry-Schneider, J. (2005a). Les adjectifs intensifs : syntaxe et sémantique. *Cahiers de Lexicologie*, 86(1):163–178. 67

Giry-Schneider, J. (2005b). Les noms épistémiques et leurs verbes supports. *Lingvisticæ Investigationes*, 27(2):219–238. 67, 114

Giry-Schneider, J. et Balibar-Mrabti, A. (1993). Classes de noms construits avec avoir. Rapport technique, LADL, Université Paris 7. 67, 82, 113

Grishman, R., Macleod, C. et Meyers, A. (1994). ComLex Syntax : Building a computational lexicon. *In Proceedings of the 15th International Conference on Computational Linguistics (CoLing'94)*, pages 268–272, Kyoto, Japon. 13, 30, 70

Gross, G. (1989). *Les constructions converses du français*. Droz, Genève, Suisse. 67, 121, 123, 127

Gross, M. (1969). Remarques sur la notion d'objet direct en français. *Langue française*, 1(1):63–73. 87

Gross, M. (1975). *Méthodes en syntaxe : Régimes des constructions complétives*. Hermann, Paris, France. 5, 7, 14, 26, 27, 38, 39, 46, 67, 96, 97, 98, 99, 100, 123, 164, 281, 288

Gross, M. (1982). Une classification des phrases «figées» du français. *Revue Québécoise de Linguistique*, 11(2):151–185. 67, 100, 131

Gross, M. (1986a). *Grammaire transformationnelle du français : Syntaxe de l'adverbe*, volume 3. ASSTRIL, Paris, France. 67, 138

Gross, M. (1986b). *Grammaire transformationnelle du français : Syntaxe du verbe*, volume 1. Cantilène, Paris, France. 33, 35

Gross, M. (1988). Adjectifs composés. *Grammaire et histoire de la grammaire*, pages 211–233. Hommage à la mémoire de Jean Stefanini. 67, 131

Gross, M. (1990). La caractérisation des adverbes dans un Lexique-Grammaire. *Langue française*, 86(1):90–102. 138, 139, 223

Gross, M. (1996). Les formes être prép x du français. *Lingvisticæ Investigationes*, (2). 67, 131

Guillet, A. et Leclère, C. (1992). *La structure des phrases simples en français : Les constructions transitives locatives*. Droz, Genève, Suisse. 39, 67, 77, 83, 100, 101, 102, 104, 106, 109, 110, 277, 293, 294, 299, 300

HAMON, O., MOSTEFA, D., AYACHE, C., PAROUBEK, P., VILNAT, A. et DE LA CLERGERIE, E. (2008). Passage : from French parser evaluation to large sized treebank. *In Proceedings of the 6th Language Resource and Evaluation Conference (LREC'08)*, Marrakech, Maroc. 5, 7, 20

HARRIS, Z. (1952). Discourse analysis. *Language*, 28(1):1–30. 92

HATHOUT, N. et NAMER, F. (1997). Génération (semi)-automatique de ressources lexicales réutilisables à grande échelle. *In Actes des 1ères JST FRANCIL, AUPELF-UREF*, Avignon, France. 80, 83, 91

HATHOUT, N. et NAMER, F. (1998). Automatic construction and validation of French large lexical resources : Reuse of verb theoretical linguistic descriptions. *In Proceedings of the 1st Language Resources and Evaluation Conference (LREC'98)*, Grenade, Espagne. 156, 157

hye HAN, C., LAVOIE, B., PALMER, M., RAMBOW, O., KITTREDGE, R., KORELSKY, T., KIM, N. et KIM, M. (2000). Handling structural divergences and recovering dropped arguments in a Korean/English machine translation system. *In AMTA '00 : Proceedings of the 4th Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, volume 1934, pages 40–53, Londres, Angleterre. Springer Verlag. 26

IDE, N. et VÉRONIS, J. (1994). MULTEXT : Multilingual text tools and corpora. *In Proceedings of the 15th International Conference on Computational Linguistics (CoLing'94)*, pages 588–592, Kyoto, Japon. 24

IOANNIDOU, K. et KYRIACOPOULOU, T. (2010). Lexique-Grammaire des verbes en grec moderne : Ambiguïtés formelles et analyse des textes. *In* POPOVIĆ, L., KRSTEV, C., VITAS, D., PAVLOVIĆ-LAŽETIĆ, G. et OBRADOVIĆ, I., éditeurs : *Actes du 29ème Colloque Lexique et Grammaire (LGC'10)*, Belgrade, Serbie. 223, 224

IOANNIDOU, K. et TOLONE, E. (2011). Construction du lexique *LGLex* à partir des tables du Lexique-Grammaire des verbes du grec moderne. *In Actes du 30ème Colloque Lexique et Grammaire (LGC'11)*, Nicosie, Chypre. À paraître. 224

JOSHI, A. K. et HOPELY, P. (1996). A parser from antiquity. *Natural Language Engineering*, 2(4):291–294. 19

JOSHI, A. K., LEVY, L. et TAKAHASHI, M. (1975). Tree Adjunct Grammars. *Journal of Computer and System Science 10*, 10(1):136–163. 23, 24, 52

KIPPER, K., DANG, H. T. et PALMER, M. (2000). Class-based construction of a verb lexicon. *In Proceedings of the 7th Conference on Artificial Intelligence (AAAI'00) and of the 12th Conference on Innovative Applications of Artificial Intelligence (IAAI'00)*, pages 691–696, Menlo Park, États-Unis. AAAI Press. 14, 30

KIPPER-SCHULER, K. (2003). *VerbNet : a broad coverage, comprehensive, verb lexicon.* Thèse de doctorat, Université de Pennsylvanie, États-Unis. 30

KLEIN, D. et MANNING, C. D. (2003). Accurate unlexicalized parsing. *In Proceedings of Association for Computational Linguistics (ACL'03)*, pages 423–430, Sapporo, Japon. 20

KUPŚĆ, A. (2007). Extraction automatique de cadres de sous-catégorisation verbale pour le français à partir d'un corpus arboré. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'07)*, Toulouse, France. 30

KYRIACOPOULOU, T. (2010). Lexique-Grammaire des verbes en grec modernes : bilan et perspectives. *Cahiers du CENTAL*, 6:181–189. Mélanges en hommage à Christian Leclère. 223

LABELLE, J. (1974). *Étude de constructions avec opérateur avoir (nominalisations et extensions).* Thèse de doctorat, LADL, Université Paris 7, France. 67, 128

LAPORTE, E. (2005). Une classe d'adjectifs de localisation. *Cahiers de Lexicologie*, 86:145–161. 67

LAPORTE, E. (2008). Exemples attestés et exemples construits dans la pratique du Lexique-Grammaire. *In* Jacques FRANÇOIS, éditeur : *Observations et manipulations en linguistique : entre concurrence et complémentarité*, volume 16 de *Mémoires de la Société de linguistique de Paris. Nouvelle série*, pages 11–32, Paris, France. Peeters. 91

LAPORTE, E. (2010). Le Lexique-Grammaire est-il exploitable pour le traitement des langues ? *Cahiers du CENTAL*, 6:207–218. Mélanges en hommage à Christian Leclère. 68, 69, 75, 77, 82, 87, 91

LAURENCE, D., BENOÏT, S. et SUSANNE, S.-A. (2006). French frozen verbal expressions : from Lexicon-Grammar tables to NLP applications. *In Actes du 25ème Colloque Lexique et Grammaire (LGC'06)*, Palerme, Italie. 47, 186

LAURENT, D., NÈGRE, S. et SÉGUÉLA, P. (2009). L'analyseur syntaxique cordial dans Passage. *In Actes de TALN'09 (session poster)*, Senlis, France. 22, 25

LECLÈRE, C. (1990). Organisation du Lexique-Grammaire des verbes français. *Langue française*, 87(1):112–122. 67, 109, 136

LEVIN, B. (1993). *English Verb Classes and Alternations : a preliminary investigation.* University of Chicago Press, Chicago, États-Unis. 14, 30, 91

MACLEOD, C., MEYERS, A., GRISHMAN, R., BARRETT, L. et REEVES, R. (1997). Designing a dictionary of derived nominals. *In Proceedings of Recent Advances in Natural Language Processing (RANLP'97)*, Tzigov Chark, Bulgarie. 14, 30

Martin, J. (2006). Mieux comprendre les méta-grammaires. Mémoire de D.E.A., Université Paris 6, France. 53

Mel'čuk, I. A., Arbatchewsky-Jumarie, N., Elnitsky, L. et Lessard, A. (1984, 1988, 1992, 1999). *Dictionnaire explicatif et combinatoire du français contemporain : Recherches lexico-sémantiques*. Presses de l'Université de Montréal, Montréal, Canada. Volumes I, II, III, IV. 26, 29

Mel'čuk, I. A. et Polguère, A. (2006). Dérivations sémantiques et collocations dans le DiCo/LAF. *Langue française*, 150(2):66–83. 29

Messiant, C. (2010). *Acquisition automatique de schémas de sous-catégorisation à partir de corpus bruts*. Thèse de doctorat, Université Paris-Nord, France. 30

Messiant, C., Namakura, T. et Voyatzi, S. (2009). La complémentarité des approches manuelle et automatique en acquisition lexicale. *In Actes de TALN'09 (session poster)*, Senlis, France. (10 pp.). 30

Meunier, A. (1981). *Nominalisations d'adjectifs par verbes supports*. Thèse de doctorat, LADL, Université Paris 7, France. 67, 82, 114, 115, 116, 123

Moliner, C. (1984). *Étude syntaxique et sémantique des adverbes de manière en -ment*. Thèse de doctorat, Université de Toulouse – Le Mirail, France. 138

Molinero, M., Sagot, B. et Nicolas, L. (2009). A morphological and syntactic wide-coverage lexicon for Spanish : The Le*ffe*. *In RANLP 2009*, Borovets, Bulgarie. 46

Molinier, C. et Levrier, F. (2000). *Grammaire des adverbes : description des formes en -ment*. Droz, Genève, Suisse. 67, 138

Paroubek, P., Robba, I., Vilnat, A. et Ayache, C. (2006). Data annotations and measures in EASy : the evaluation campaign for parsers of French. *In Proceedings of the 5th Language Resource and Evaluation Conference (LREC'06)*, pages 315–320, Gênes, Italie. 20, 21

Paroubek, P., Robba, I., Vilnat, A. et Ayache, C. (2008). EASy, evaluation of parsers of French : what are the results ? *In Proceedings of the 6th Language Resource and Evaluation Conference (LREC'08)*, Marrakech, Maroc. 20

Paumier, S. (2001). Some remarks on the application of a Lexicon-Grammar. *Lingvisticæ Investigationes*, 24(2):245–256. 84

Paumier, S. (2003). *De la reconnaissance de formes linguistiques à l'analyse syntaxique*. Thèse de doctorat, Université Paris-Est Marne-la-Vallée, France. 77, 83, 105, 156, 161, 200

Picabia, L. (1978). *Les constructions adjectivales en français : systématique transformationnelle*. Droz, Genève, Suisse. 67

Pivaut, L. (1989). *Verbes supports et vocabulaire technique : sport, musique et activités intelectuelles*. Thèse de doctorat, LADL, Université Paris 7, France. 67, 129

Rogers, J. et Vijay-Shanker, K. (1992). Reasoning with descriptions of trees. *In Proceedings of Association for Computational Linguistics (ACL'92)*, pages 72–80, Louvain, Belgique. 54

Roussanaly, A., Crabbé, B. et Perrin, J. (2005). Premier bilan de la participation du loria à la campagne d'évaluation EASy. *In Actes de l'atelier EASy de TALN'05*, pages 49–52, Dourdan, France. 22, 24

Ruwet, N. (1972). *Théorie syntaxique et syntaxe du français*. Seuil, Paris, France. 294

Sagot, B. (2005). Automatic acquisition of a Slovak lexicon from a raw corpus. *In Text, Speech and Dialogue, 8th International Conference, TSD 2005, Karlovy Vary, Czech Republic, September 12-15, 2005, Proceedings*, volume 3658 de *Lecture Notes in Artificial Intelligence (LNAI)*, pages 156–163. Springer Verlag. ISBN 978-3-540-28789-6. 46

Sagot, B. (2006). *Analyse automatique du francais : lexiques, formalismes, analyseurs*. Thèse de doctorat, Université Paris 7, France. 46

Sagot, B. (2010). The Le*fff*, a freely available and large-coverage morphological and syntactic lexicon for French. *In Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10)*, La Valette, Malte. 5, 7, 14, 27, 45, 47, 49, 257

Sagot, B. et Boullier, P. (2008). SxPipe 2 : architecture pour le traitement présyntaxique de corpus bruts. *Traitement Automatique des Langues (T.A.L.)*, 49(2):155–188. 23

Sagot, B. et Danlos, L. (2007). Améliorer un lexique syntaxique à l'aide des tables du Lexique-Grammaire : Constructions impersonnelles. *Cahiers du CENTAL*, 5:107–126. 14, 27, 45, 46, 49, 156

Sagot, B. et de La Clergerie, E. (2008). Fouille d'erreurs sur les sorties d'analyseurs syntaxiques. *Traitement Automatique des Langues (T.A.L.)*, 49(1):41–60. 46, 94, 207, 208, 209, 226, 227

Sagot, B. et Fort, K. (2007). Améliorer un lexique syntaxique à l'aide des tables du Lexique-Grammaire : adverbes en *-ment*. *In Actes du 26ème Colloque Lexique et Grammaire (LGC'07)*, pages 145–152, Bonifacio, France. 47, 140

Sagot, B. et Fort, K. (2009). Description et analyse des verbes désadjectivaux et dénominaux en *-ifier* et *-iser*. *Arena Romanistica, Journal of Romance Studies*, 4:102–110. ISSN 1473-3536. 47

SAGOT, B. et TOLONE, E. (2009a). Exploitation des tables du Lexique-Grammaire pour l'analyse syntaxique automatique. *Arena Romanistica, Journal of Romance Studies*, 4:302–312. ISSN 1473-3536. 94, 205

SAGOT, B. et TOLONE, E. (2009b). Intégrer les tables du Lexique-Grammaire à un analyseur syntaxique robuste à grande échelle. *In Actes de TALN'09 (session poster)*, Senlis, France. (10 pp.). 94, 175, 205

SALKOFF, M. et VALLI, A. (2005). A dictionary of French verbal complementation. *In Proceedings of the 2nd Language and Technology Conference (LTC'05)*, Poznań, Pologne. 29

SCHABES, Y. et WATERS, R. (1995). Tree Insertion Grammar : a cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Fuzzy Sets and Systems*, 76(3):309–317. 56

SFETSIOU, V. (2007). *Noms prédicatifs : méthode d'analyse pour des applications eléctroniques*. Thèse de doctorat, Université Aristote de Thessalonique, Grèce. 223

SURDEANU, M., HARABAGIU, S. M., WILLIAMS, J. et AARSETH, P. (2003). Using predicate-argument structures for information extraction. *In Proceedings of Association for Computational Linguistics (ACL'03)*, pages 8–15. 26

THOMASSET, F. et DE LA CLERGERIE, E. (2005). Comment obtenir plus des méta-grammaires. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'05)*, Dourdan, France. 5, 7, 14, 22, 23, 53, 56, 57, 60, 175

TOLONE, E. (2009). Les tables du Lexique-Grammaire au format TAL. *In Actes de MajecSTIC 2009*, Avignon, France. (8 pp.). 20, 65, 81, 88, 94, 97, 155, 195, 243

TOLONE, E. (2010). Constructions définitoires des tables du Lexique-Grammaire : Le cas des complétives et des infinitives. *In Actes de MajecSTIC 2010 (session poster)*, Bordeaux, France. (4 pp.). 96

TOLONE, E. (2011). Conversión de las tablas del Léxico-Gramática del francés en el léxico *LGLex*. 2nd Argentinian Workshop on Natural Language Processing (WNLP'11). À paraître. 129, 225

TOLONE, E. et SAGOT, B. (2011). Using Lexicon-Grammar tables for French verbs in a large-coverage parser. *In* VETULANI, Z., éditeur : *Human Language Technology, Forth Language and Technology Conference, LTC 2009, Poznań, Poland, November 2009, Revised Selected Papers*, Lecture Notes in Artificial Intelligence (LNAI). Springer Verlag. À paraître. 94, 175, 205, 257

TOLONE, E., SAGOT, B. et DE LA CLERGERIE, E. (2011). Évaluation de lexiques syntaxiques par leur intégration dans l'analyseur syntaxique FRMG. *In Actes du 30ème Colloque Lexique et Grammaire (LGC'11)*, Nicosie, Chypre. À paraître. 209, 225, 226

TOLONE, E. et VOYATZI, S. (2011). Extending the adverbial coverage of a NLP oriented resource for French. *In Proceedings of IJCNLP'11 (poster)*, Chiang Mai, Thaïlande. À paraître. 140, 166, 225

TOLONE, E., VOYATZI, S. et LECLÈRE, C. (2010). Constructions définitoires des tables du Lexique-Grammaire. *In* POPOVIĆ, L., KRSTEV, C., VITAS, D., PAVLOVIĆ-LAŽETIĆ, G. et OBRADOVIĆ, I., éditeurs : *Actes du 29ème Colloque Lexique et Grammaire (LGC'10)*, pages 321–331, Belgrade, Serbie. 100, 108, 112, 131, 138, 139

VAN DEN EYNDE, K. et MERTENS, P. (2003). La valence : l'approche pronominale et son application au lexique verbal. *Journal of French Language Studies*, 13:63–104. 27

VAN DEN EYNDE, K. et MERTENS, P. (2006). *Le dictionnaire de valence* DICOVALENCE *: manuel d'utilisation*. http ://bach.arts.kuleuven.be/dicovalence/manuel_061117.pdf. 27, 28, 46, 49, 180

VANRULLEN, T., BLACHE, P. et BALFOURIER, J.-M. (2006). Constraint-based parsing as an efficient solution : Results from the parsing evaluation campaign EASy. *In Proceedings of the 5th Language Resource and Evaluation Conference (LREC'06)*, Gênes, Italie. 22, 25

VANRULLEN, T., BLACHE, P., PORTES, C., RAUZY, S., MAEYHIEUX, J.-F., GUÉNOT, M.-L., BALFOURIER, M.-L. et BELLENGIER, J.-M. (2005). Une plateforme pour l'acquisition, la maintenance et la validation de ressources lexicales. *In Actes de la Conférence sur le Traitement Automatique des Langues Naturelles (TALN'05)*, Dourdan, France. 29

VERONIS, J. (1998). Multext-lexicons, a set of electronic lexicons for european languages. CD-ROM distributed by ELRA/ELDA. 47

VIJAY-SHANKER, K. (1987). *A study of Tree Adjoining Grammar*. Thèse de doctorat, Université de Pennsylvanie, États-Unis. 24, 52

VIVÈS, R. (1983). *Avoir, prendre, perdre : constructions à verbe support et extensions aspectuelles*. Thèse de doctorat, LADL, Université Paris 7, France. 67, 74, 123, 126

VOYATZI, S. (2006). *Description morpho-syntaxique et sémantique des adverbes figés en vue d'un système d'analyse automatique des textes grecs*. Thèse de doctorat, Université Paris-Est Marne-la-Vallée, France. 223

VOYATZI, S. et KAKOYIANNI-DOA, F. (2010). The electronic dictionary of Modern Greek adverbs. *In 30th Annual Meeting of the Department of Linguistics, Faculty of Philology, Aristotle University of Thessaloniki*, Studies in Greek Linguistics, pages 149–161. 223

Cinquième partie

Annexes



Format du lexique *LGLex*

The *LGLex* lexicon is a syntactic lexicon of French verbs, predicative nouns, frozen expressions and adverbs generated by the *LGExtract* tool (Constant et Tolone, 2010) from the Lexicon-Grammar tables under LGPL-LR license. It is available both in text and XML format.
For more details about the modifications of tables, see (Tolone, 2009).

Remark:
Some tables of nouns have also morphologically derived verb:
f1a,f1b,f1c,f1d,f1r,f2a,f2b,f2c,f21,f3,f4,f5,f9,ad,dr1,drc,es,fr1,fs1,is1
Others have morphologically derived adjective:
an01,an02,an03,an04,an05,an06,ansy,es
In addition, the entries of the adverb table peco are also predicative adjectives, while those of the adverb tables ppco and pvco are also frozen expressions.
These additional entries, in addition to being mentioned in the complete lexicon, were extracted in the directory *extra-lexicon/*.

A.1 Description of the content of the *LGLex* lexicon

In XML format, you have the element <**syn-lexicon**>:
Each entry is delimited by the tag <**entry**> with an attribute id which is the entry identifier generated by concatenating its grammatical category, its class (or table) it comes from and the index of the entry in the table. For instance, in <**entry id="V_33_24" status="completed"**>, the identifier is **V_33_24**, which correspond to the 24th entry in the verb class 33. The status can be **completed** for a fully coded entry, **to complete** for an entry that has at least one property uncoded, or **to encode** for an entry that has

at least half of its properties uncoded.
Note: We call here a property encoded, a property present in the table and coded ∼. This ignores the coding O in the table of classes also means that the property must be encoded, but not listed in the table.
Then, information is gathered in four different tags:
- <**lexical-info**>: lemma and lexical information;
- <**arguments**>: arguments and their nature;
- <**all-construction**>: accepted constructions;
- <**example**>: an illustrative example.

A.1.1 Lexical information

<**lexical-info**> contains lexical information corresponding to the entry and has an attribute cat indicating its category (**verb**, **noun** for predicative noun, **adj** for predicative adjective, **expr** for frozen expression and **adverb**):

For verbs ONLY:
- <**lexical-value**> includes <**lemma**>, the value of which is the lemma of the entry, that can be completed with preverbal pronouns like *se*, *y*, *en*, *le*, *la*, *les* and/or the negation *ne ... pas* when they are obligatory in this entry (tags <**ppvse**>, <**ppvy**>, <**ppven**>, <**ppvle**>, <**ppvla**>, <**ppvles**>, <**neg**> with the attribute value **true**);
- <**aux**> indicates the possible auxiliaries accepted by this entry. It contains a list including the tags <**avoir**> or <**être**> with value **true**;
- <**traduction**> contains for some entries the translation of the lemma in English;

For nouns, adjective, frozen expressions and adverbs:
- <**lexical-value**> is composed of element <**complete**>, the value of which is the whole entry (it can be multiword) and the elements <**noun1**>, <**adj1**>, <**det2**>, <**noun2**> (for nouns), containing the values of its different components. The morphologically derived adjective (resp. verb) might also be indicated in tag <**adjassoc**> (resp. <**verbassoc**>). Complete list of elements for all categories: <**adj**>, <**adj1**>, <**adv**>, <**adv1**>, <**c**>, <**c0**>, <**c02**>, <**c1**>, <**c12**>, <**c2**>, <**c3**>, <**cc**>, <**cv**>, <**conj**>, <**conj2**>, <**conjcoord**>, <**conjsub**>, <**det**>, <**det0**>, <**det02**>, <**det1**>, <**det12**>, <**det2**>, <**det3**>, <**detc**>, <**detv**>, <**ilya**>, <**modif**>, <**MPA**>, <**noun1**>, <**noun2**>, <**nv**>, <**poss1**>, <**prep**>, <**prep0**>, <**prep1**>, <**prep2**>, <**prep3**>, <**prepdetv**>, <**prepc**>, <**prepv**>, <**verb**>, <**verb2**>, <**comme**>, <**ce**>, <**il**>, <**ca**> and also <**adjassoc**>, <**advassoc**>, <**nounassoc**>, <**ppvassos**>, <**ppvseassoc**>, <**ppvenassoc**>, <**ppvyassoc**>, <**ppvnegassoc**>, <**verbassoc**>;

For nouns and frozen expressions:
- <**Vsup**> describes the support verbs associated with the deverbal noun(or the frozen expression made of an adjective or an adverb and a support verb). It contains

<**cat**>, which has always the value **verb** and a list of <**value**>, representing the possible lexical values of the support verb part of basic construction of the entry;

For nouns ONLY:

- <**Vconv**> is constructed in the same manner of <**Vsup**> and refers to a second list of lexical values for the converse support verb Vconv as it can appear in some converse constructions (e.g. N1 Vconv Det N à N0);
- <**det-modif-list**> includes a list of <**determiner-modifier**> which indicates the distribution of determiners with possible modifiers for the noun. <**determiner-modifier**> has a tag <**det**>, which indicates the possible determiners separated by + (the value <**E**> stands for the absence of determiner). Tag <**modifier**> indicates whether the noun accept a modifier with the defined determiners. This can be completed with <**value-modif**>, which indicates the possible modifiers separated by +;

For verbs and nouns:

- <**prepositions**> contains a sequence of <**preposition**> with an attribute id, corresponding to the number of the argument it introduces in the elementary construction (0 for subject, 1 for the first argument, 2 for for the second one, ...). In <**preposition**>, <**prep**> represents the different lexical values of the preposition. For instance, the deverbal noun *allergie* (*allergy*) in table an01 uses support verb avoir (have) and enters the elementary sentence N0 Vsup Det N Prép N1. The preposition associated with argument 1 is specified as follows:

```
<prepositions>
      <preposition id="1">
        <prep value="à" />
      </preposition>
</prepositions>
```

  In the first complement (Prép N1), the preposition has the value *à*: e.g., *Léa a une certaine allergie* **à la poussière** (*Lea is allergic* **to dust**);
- <**locatifs**> defines the locative preposition distributions. It contains a sequence of <**locatif**> with an attribute id, referring to the number of the associated argument. Element <**loc**> also has a list of <**prep**>, representing the possible lexical values of the locative prepositions.

A.1.2 Distribution of arguments

<**arguments**> describes the distribution of the different arguments (subject and complements) of the entry. It includes a set of <**constituent**> having an attribute pos, that indicates the number of the associated argument in the elementary sentence that the entry enters. A constituent is a list of <**component**>, each of them having the following elements:

- <**cat**> which has an attribute specifying its syntactic nature: **NP** for noun phrase,

inf for infinitive (V-inf W), **comp** for a complementizer phrase (Qu P), **leFaitComp** for the noun phrase le fait que P, **siPOuSiP** for the specific complementizer phrase si P ou si P and **adj** for an adjective;
- Various semantic features <**hum**> (human), <**nothum**> (non human), <**plobl**> (obligatory plural), <**npr**> (proper noun), <**abst**> (abstract) and <**conc**> (concrete) with the value **true** when they are verified;

For verbs ONLY:
- Other semantic features are possible for verbs: <**source**>, <**destination**>, <**beneficiaire**> (beneficiary), <**detrimentaire**> (detrimental), <**apparition**> (appearance), <**disparition**> (disappearance), <**mesure**> (measure), <**prix**> (price); and for nouns: <**coll**> (collective noun), <**plur**> (plural);

There also exist several other optional features:
- <**mood**> with an attribute indicating the mood of the complementizer phrase (**ind** for indicative and **subj** for subjunctive);
- <**contr**> with an attribute indicating the number of the argument that controls the infinitive. In the previous example, the distribution of the argument at position 1 (i.e., the first complement Prép N1) is described as follows:

```
<arguments>
   ...
  <constituent pos="1">
    <component>
      <cat value="inf" />
      <contr value="0" />
    </component>
    <component>
      <cat value="ceComp" />
      <mood value="ind" />
    </component>
    <component>
      <cat value="ceComp" />
      <mood value="subj" />
    </component>
    <component>
      <cat value="NP" />
      <nothum value="true" />
    </component>
  </constituent>
  ...
</arguments>
```

The complement Prép N1 can be:
- an infinitive controlled by argument 0, i.e., the subject N0: e.g., *Léa a une allergie*

à travailler (*Lea is allergic* **to work**):
- a complementizer phrase in the indicative: e.g, *Léa a une allergie* **à ce que nous voyageons** (*Lea is allergic* **to the fact that we travel**);
- a complementizer phrase in the subjective: e.g., *Léa a une allergie* **à ce qu'il fasse beau** (*Lea is allergic* **to shine**);
- a noun phrase (see previous example).

For verbs ONLY:
- <**origin**> contains the list <**orig**>, which indicates the complete name of the columns in the table, that has been used to define the distribution;
- <**introd-prep**> contains a sequence of <**prep**>, providing the prepositions introducing the argument and indicating their lexical values;
- <**introd-loc**> contains a sequence of <**loc**>, providing the locative prepositions that introduce the argument and indicating their lexical values.

A.1.3 Constructions

<**all-constructions**> list the different constructions that are accepted by the entry:
- <**absolute-constructions**> includes a list of absolute <**construction**>, which values are the titles of the columns entirely specifying the accepted construction with all its constituents. For instance, in the construction N0 V N1 Prép N2, N0 represents the subject, V indicates the verbal entry, N1 is the first complement and Prép N2 is the second one introduced by the preposition Prép. The construction title is preceded by the string **o::** if the column associated with the construction has been coded + in the table (**o**) or **true::** if it is a constant + in the table of classes (**true**). In this last case, the property is verified by all the entries of the table. Such properties are the definitional properties of each classes, including the base construction;
- <**relative-constructions**> contains the sequence of all relative <**construction**>, which values are the column titles of all columns specifying the name of the transformations that are applied on the base construction (e.g., [passif par]);

For nouns ONLY:
- <**reductionsGN**> describing reductions of the base sentence construction into another construction of another syntactic category, in the present case, noun phrase. Reductions are described with a list of <**construction**>, the value of which are the titles of the columns specifying the reduction construction (e.g., le N entre N0 et N1);
- <**verbales**> includes a list of verbal <**construction**>, which values are the titles of the columns specifying the construction which are accepted by the corresponding verbal entry (<**verbassoc**>);

For adverbs ONLY:
- <**structureAdv**> contains the base structure of the multiword adverb (e.g., Prép Det Adj C) and also its variants into a list of <**construction**> (e.g., Prép Det C). This structure represents Adv in the absolute and relative constructions (e.g., the ab-

solute construction N0 V Adv W can be writted N0 V Prép Det Adj C W). For simple adverbs, the structure is not indicated because it's represented directly by Adv in the constructions.

A.1.4 Example

<**example**> illustrates the entry (solely for verbs and nouns):
The value of <**example**> is an example of sentence with the entry.
For verbs, all entries contains an example, whereas, for nouns, only a selection of nouns contains an example.

A.1.5 The *LGLex* lexicon in text format

An entry described in text format includes the same information and is represented as follows:

```
ID=category_tableNumber_entryNumber;status=...
lexical-info=[...]
args=(...)
all-constructions=[absolute=(...),
                   relative=(...)]
example=[...]
```

A.2 Examples

A.2.1 A verbal example of *LGLex* lexicon in the XML format

The verb *candidater* (*to apply*) in the table 33:

```
<entry id="V_33_24" status="completed">
  <lexical-info cat="verb">
    <lexical-value>
      <lemma value="candidater" />
    </lexical-value>
      <aux>
        <avoir value="true" />
      </aux>
    <locatifs />
    <prepositions />
  </lexical-info>
  <arguments>
    <constituent pos="0">
      <component>
        <cat value="NP" />
        <hum value="true" />
```

```
        <origin>
          <orig value="N0 =: Nhum" />
        </origin>
        <introd-prep />
        <introd-loc />
      </component>
    </constituent>
    <constituent pos="1">
      <component>
        <cat value="NP" />
        <nothum value="true" />
        <origin>
          <orig value="N1 =: N-hum" />
        </origin>
        <introd-prep />
        <introd-loc />
      </component>
    </constituent>
  </arguments>
  <all-constructions>
    <absolute-constructions>
      <construction value="true::N0 V à N1" />
    </absolute-constructions>
    <relative-constructions>
      <construction value="[extrap]" />
      <construction value="Ppv =: y" />
    </relative-constructions>
  </all-constructions>
  <example>
    <example value="Max a candidaté à un poste" />
  </example>
</entry>
```

A.2.2 Same example in text format

```
ID=V_33_24;status=completed
lexical-info=[cat="verb",verb=[lemma="candidater"],
              aux-list=(avoir="true"),prepositions=(),locatifs=()]
args=(const=[pos="0",
             dist=(comp=[cat="NP",hum="true",
                         origin=(orig="N0 =: Nhum"),
                         introd-prep=(),introd-loc=()])],
      const=[pos="1",
             dist=(comp=[cat="NP",nothum="true",
                         origin=(orig="N1 =: N-hum"),
                         introd-prep=(),introd-loc=()])])
```

```
all-constructions=[absolute=(construction="true::N0 V à N1"),
                   relative=(construction="[extrap]",
                             construction="Ppv =: y")]
example=[example="Max a candidaté à un poste"]
```

A.2.3 A nominal example of *LGLex* lexicon in XML format

The noun *bise* in the table fnan, with support verb *faire*: *faire la bise* (*to kiss on the cheeks*):

```
<entry id="N_fnan_18" status="completed">
  <lexical-info cat="noun">
    <lexical-value>
      <complete value="bise" />
      <noun1 value="bise" />
    </lexical-value>
    <Vsup>
      <cat value="verb" />
      <value value="faire" />
    </Vsup>
    <det-modif-list>
      <determiner-modifier>
        <det value="un+une" />
        <modif value="false" />
      </determiner-modifier>
      <determiner-modifier>
        <det value="un+une" />
        <modif value="true" />
      </determiner-modifier>
      <determiner-modifier>
        <det value="la" />
        <modif value="false" />
      </determiner-modifier>
    </det-modif-list>
    <prepositions />
  </lexical-info>
  <arguments>
    <constituent pos="0">
      <component>
        <hum value="true" />
        <cat value="NP" />
      </component>
    </constituent>
    <constituent pos="1">
      <component>
        <hum value="true" />
```

```
      <cat value="NP" />
    </component>
  </constituent>
</arguments>
<all-constructions>
  <absolute-constructions>
    <construction value="true::N0 Vsup Det N à N1" />
    <construction value="true::N0 Vsup Det N" />
  </absolute-constructions>
  <relative-constructions />
  <reductions />
</all-constructions>
<example>
  <example value="" />
</example>
</entry>
```

A.2.4 Same example in text format

```
ID=N_fnan_18;status=completed
lexical-info=[cat="noun",
              Vsup=[cat="verb",list=(value="faire")],
              noun=[notperm=[complete="bise"],noun1="bise"]],
              detN=[list-det-modif=(det-modif=[det="un+une",modif="false"],
                                    det-modif=[det="un+une",modif="true"],
                                    det-modif=[det="la",modif="false"]),
              prepositions=()]
args=(const=[pos="0",
             dist=(comp=[hum="true",cat="NP"])],
      const=[pos="1",
             dist=(comp=[hum="true",cat="NP"])])
all-constructions=[absolute=(construction="true::N0 Vsup Det N à N1",
                             construction="true::N0 Vsup Det N"),
                   relative=(),
                   verbales=(),
                   reductionsGN=()]
example=[example=]
```

B
Génération du lexique *LGLex* à l'aide de *LGExtract*

L'utilisation de l'outil *LGExtract* pour générer le lexique *LGLex* est :

```
java fr.umlv.lgextract.LGExtract --script <script> --tableDirPath
<tableDirPath> --headerColumn <headerColumn> --headerRow <headerRow> [--debug]
[--tdtValues <tdtValues>] [--values <values>] [--output <output>] <tdt>
```

avec les paramètres suivants :

`--script <script>`

Chemin du script (le script doit être encodé en UTF-8)

`--tableDirPath <tableDirPath>`

Chemin du répertoire contenant les tables

`--headerColumn <headerColumn>`

Numéro de colonne où commencent les en-têtes de propriétés dans la table des classes (la première colonne est **0**)

`--headerRow <headerRow>`

Numéro de ligne où commencent les en-têtes de tables dans la tables des classes (la première ligne est **0**)

`[--debug]`

Indique que le programme est lancé en mode debug (si n'est pas présent, il est lancé en mode normal)

`[--tdtValues <tdtValues>]`

Chemin du fichier contenant les interprétations des valeurs dans la table des classes. Par défaut : interprétation de la valeur du Lexique-Grammaire standard

`[--values <values>]`

Chemin du fichier contenant les interprétations des valeurs dans les tables du Lexique-Grammaire. Par défaut : interprétation de la valeur du Lexique-Grammaire standard

`[--output <output>]`

Types de sorties : texte (`txt`) ou XML (`xml`). Valeur par défaut : `txt`

`[--tableFormat <tableFormat>]`

Format des tables : Excel (`xls`) ou CSV (`csv`). Valeur par défaut : `csv`

`<tdt>`

Liste des tables des classes utilisées

Les bibliotèques externes utilisées sont :
- Tatoo (http://tatoo.univ-mlv.fr/) : utilisé pour parcourir le script de configuration `[tatoo-runtime.jar]`
- Velocity-dep (http://mvnrepository.com/artifact/velocity/velocity-dep) : utilisé par Tatoo `[velocity-dep-1.4.jar]`
- JDom (http://www.jdom.org/) : utilisé pour généré la sortie XML `[jdom.jar]`
- JExcelApi (http://jexcelapi.sourceforge.net/) : utilisé pour parcourir les fichiers Excel `[jxl.jar]`
- JSAP (http://martiansoftware.com/jsap/) : utilisé pour parcourir la ligne de commande des arguments `[JSAP-2.1.jar]`

Les fichiers d'archive *.jar* de ces bibliotèques sont inclus dans le répertoire *jar*. Ils doivent être ajoutés à la variable CLASSPATH pour utiliser *LGExtract*.
Exemple :

```
export CLASSPATH=$CLASSPATH:classes:jar/tatoo-runtime.jar:jar/
      velocity-dep-1.4.jar:jar/jxl.jar:jar/JSAP-2.1.jar:jar/jdom.jar
```

Détaillons les différents fichiers utilisés avec l'outil *LGExtract* :
- 8 exécutables prédéfinis permettent de lancer l'outil *LGExtract* sur une catégorie donnée (verbes, noms prédicatifs, expressions figées ou adverbes) et de créer le lexique *LGLex* au format texte ou XML.
  Notons que l'utilisation de ces exécutables requiert la création de la variable d'environnement `$TABLESPATH` dans *.bashrc*, indiquant le chemin du répertoire principal contenant toutes les données (tables) et l'outil *LGExtract* (en adaptant le path du répertoire `tables/`) :

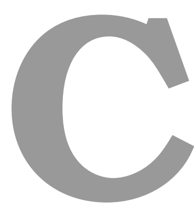
```
export TABLESPATH="$HOME/tables"
```
Voici quelques exemples :
```
./launch (affiche l'aide)
./launch_verbes > $TABLESPATH/lglex/verbes-lglex.txt
./launch_noms-predicatifs > $TABLESPATH/lglex/noms-predicatifs-lglex.txt
./launch_figees > $TABLESPATH/lglex/figees-lglex.txt
./launch_adverbes > $TABLESPATH/lglex/adverbes-lglex.txt
./launch_verbes_xml > $TABLESPATH/lglex/verbes-lglex.xml
./launch_noms-predicatifs_xml
  > $TABLESPATH/lglex/noms-predicatifs-lglex.xml
./launch_figees_xml > $TABLESPATH/lglex/figees-lglex.xml
./launch_adverbes_xml > $TABLESPATH/lglex/adverbes-lglex.xml
```
À la fin de la compilation, un message indique le nombre d'entrées créées à partir de combien de tables, ainsi que le nombre d'entrées ayant le statut **to complete** et **to encode** (cf. section 7.3) ;

– 4 scripts de configuration (avec l'extension *.lg*), un par catégorie (*lgc_verbes.lg*, *lgc_noms-predicatif.lg*, *lgc_figees.lg* et *lgc_adverbes.lg*). Les fichiers de configuration sont encodé en UTF-8 ;

– 2 fichiers (facultatifs) spécifiant la signification des différentes valeurs dans les tables et dans la table des classes.
Leur format est : symbole **s** suivi d'un espace, suivi de la signification **true** ou **false**.
tables-values.txt contient :
```
+ true
- false
~ false
```
tablesOfClasses-values.txt contient :
```
+ true
- false
/ false
?+ false
?- false
? false
0 false
```
Des symboles tels que le chiffre 0 au lieu de la lettre O, ou tout autre erreur possible, peuvent être ajoutés ;

– 2 fichiers supplémentaires utilisés lors de la création des tables des classes et des scripts de configuration, mais non utiles pour la génération du lexique *LGLex* :
1 exécutable `makeTableOfClasses` permettant de construire la table des classes au format Excel.
L'utilisation est :
```
java LGExtractTableOfClasses <dirpath of tables> <output xls file>
```
Voici un exemple :
```
./makeTableOfTables ../verbes tdt-verbes.xls
```

1 script Perl `list2code.pl` permettant de générer automatiquement du code *.lg* à partir d'un fichier *.txt* contenant des constructions (une par ligne).



Format du lexique *LGLex*-Le*fff*

The *LGLex*-Le*fff* lexicon is a syntactic lexicon of French verbs and predicative nouns converted in the Le*fff* format (Tolone et Sagot, 2011) from the *LGLex* lexicon.
For more details about the Le*fff* lexicon and the framework it is based on, named Alexina, see Sagot (2010).

C.1 Description of the intensional Le*fff* format

Each entry in the intensional lexicon corresponds to a unique meaning of the corresponding lemma. It contains the following information:

- an entry identifier generated by concatenating its grammatical category, its class (or table) it comes from and the index of the entry in the table;
- a morphological class, which defines the patterns that build its inflected forms, using inflection classes from the Le*fff*;
- a category (or part-of-speech);
- the initial sub-categorization frame;
- additional syntactic information represented by macros;
- the list of possible redistributions;
- an example of sentence with the entry;
- a comment after the # sign.

For example, the intensional entry in the *LGLex*-Le*fff* lexicon for the French lemma *clouer___V_36SL_28* (*to nail*) is as follows:

```
clouer___V_36SL_28 v-er:std
                   100;Lemma;v;
```

```
<Suj:cln|sn,Obj:sn,Loc:(avec-sn|et-sn|à-sn|sur-sn)>;
cat=v;
%actif,%passif,%ppp_employé_comme_adj
Ex.: Max a cloué cette planche(avec+contre+sur)celle-là
# BASE CONSTR = N0 V N1 Loc N2 (N0 V N1 et N2 ;
  N0 V N1 Prép N2 ; N0 V N1hum Loc N2abs ; N0 V N1)
  [[passif par]] <> ; orig base constr = N0 V N1 Loc N2
```

It describes a transitive entry with the following information:

– its identifier is *V_36SL_28*, which correspond to the 28th entry in the verb class 36SL;
– its morphological class is *v-er:std*, the class of standard first-conjugation verbs (ending *-er*);
– its semantic predicate can be represented by the *Lemma* as is, i.e., *clouer*;
– its category is verb (*v*);
– it has three arguments canonically realized by the syntactic functions *Suj* (subject), *Obj* (direct object) and *Loc* (locative argument); each syntactic function is associated with a list of possible realizations, but the *Loc* is optional as shown by the brackets;
– it allows for three different redistributions: *%active*, *%passive*, and *%ppp_employé_comme_adj*;
– its example is *Max a cloué cette planche(avec+contre+sur)celle-là* (*Max has nailed this board(with+by+on)that one*).

C.2 Description of the extensional Lefff format

The compilation process builds one extensional entry for each inflected form and each compatible redistribution, by inflecting the lemma according to the definition of its morphological class and by applying the formalized definitions of these redistributions. For example, the only inflected forms of *clouer* that are compatible with the passive redistribution are the past participle forms. The extensional passive entry for *cloués* (*nailed*) is the following (*Kmp* is the morphological tag for past participle masculine plural forms):

```
cloués 100 v [pred="clouer___V_36SL_28__1
             <Suj:sn,Loc:(avec-sn|et-sn|sur-sn|à-sn),Obl2:(par-sn)>",
             @passive,@pers,cat=v,@Kmp]
             clouer___V_36SL_28__1 PastParticiple Kmp %passif
```

The original direct object (*Obj*) has been transformed into the passive Subject, and an optional Agent (*Obl2*) realized by a noun phrase preceded by a preposition (*par-sn*) was added.

C.3 Characteristics of the *LGLex*-Le*fff* lexicon

C.3.1 Syntactic functions

For verbs, the Le*fff* format uses the following syntactic functions:

- *Suj* for subject: cliticization with the nominative clitic;
- *Obj* for direct object: cliticization with the accusative clitic, commutable with *ceci/cela* (*this/that*), impacted by passivization when it is possible;
- *Objà* for indirect object canonically introduced by the preposition *à*: commutable with *à+non-clitic pronoun* but not with *ici* (*here*) or *là(-bas)* (*there*), may be cliticizable into the dative clitic or *y*;
- *Objde* for indirect object introduced by the preposition *de*: cliticization with *en*, not commutable with *d'ici* (*from here*) or *de là* (*from there*);
- *Loc* for locative argument: commutable with *ici* (*here*) or *là(-bas)* (*there*), cliticizable with *y*: e.g., *à Paris* in *Pierre va* **à Paris** (*Peter goes* **to Paris**);
- *Dloc* for delocative argument: commutable with *d'ici* (*from here*) or *de là* (*from there*), cliticizable with *en*: e.g., *de Paris* in *Pierre vient* **de Paris** (*Peter comes* **from Paris**);
- *Att* for (subject, object or *à*-object) attribute and pseudo-object: e.g., *3 euros* in *J'ai acheté ceci* **3 euros** (*I bought this* **3 euros**);
- *Obl* and *Obl2* for other oblique arguments (non-cliticizable); *Obl2* is used for verbs with two oblique arguments, such as *plaider auprès de quelqu'un en faveur de quelqu'un d'autre* (*to plead in front of somebody for somebody else*).

For predicative nouns, that can be headed by a support verb, the same set of functions are used.

C.3.2 Realizations

For verbs and nouns, possible realizations are threefold:

- clitic pronouns: *cln* for nominative clitic: e.g., *il* in **Il** *donne ce livre à Marie* (**He** *gives this book to Mary*), *cla* for accusative clitic: e.g., *le* in *Il* **le** *donne à Marie* (*He gives* **it** *to Mary*), *cld* for dative clitic: e.g., *lui* in *Il* **lui** *donne ce livre* (*He gives* **her** *this book*), *y*: e.g., *Max* **y** *va* (*Max goes* **there**), *en*: e.g., *Max* **en** *mange* (*Max eats* **it**);
- direct phrases: *sn* for noun phrase: e.g., *La belle dame* in **La belle dame** *arrive* (**The beautiful lady** *arrives*), *sa* for adjectival phrase: e.g., *verte* in *La robe est* **verte** (*The dress is* **green**), *sinf* for infinitive clause: e.g., *dire aurevoir* in *Pierre est parti* **dire aurevoir** (*Peter left* **to say goodbye**), *scompl* for completive clause: e.g., *que Marie est belle* in *Pierre dit* **que Marie est belle** (*Peter says* **that Marie is beautiful**), *qcompl* for interrogative clause: e.g., *combien il gagne* in *Pierre dit* **combien il gagne** (*Peter said* **how much he earns**);
- prepositional phrases: a direct phrase introduced by a preposition (e.g., *à-sn*, *de-scompl*, *pour-sinf*).

C.3.3 Redistributions

For verbs, the inventory of possible redistributions is the following:
- *%actif*, a dummy redistribution that has almost no effect on the initial subcategorization information;
- *%passif* for the standard passive in *par*: e.g., *Jean assistait Max depuis des années → Max était assisté par Jean depuis des années* (*John has assisted Max for years → Max has been assisted by John for years*);
- *%passif_de* for the passive in *de*: e.g., *Marie aime Pierre → Pierre est aimé de Marie* (*Mary loves Peter → Peter is loved by Mary*);
- *%actif_impersonnel* for active impersonal constructions with inverted subject, if any: e.g., *Un accident est arrivé à Jean → Il est arrivé un accident à Jean* (*An accident happened to John*);
- *%passif_impersonnel* for passive impersonal constructions with inverted subject, if any: e.g., *Cette nouvelle information clarifie pourquoi Max est fâché → Il est clarifié par cette nouvelle information pourquoi Max est fâché* (*This new information clarifies why Max is angry → It is clarified by this new information why Max is angry*);
- *%ppp_employé_comme_adj* for past participles used as adjectives: e.g., *Marie s'assoie → Marie est assise* (*Mary is sitting → Mary is seated*).

Predicative nouns only use the *%default* redistribution that builds a final subcategorization frame which is identical to the initial one. In addition, they can have the *%passif* redistribution.

C.3.4 Macros

For verbs and nouns, macros represents additional syntactic information such as:
- control: for instance, *@CtrlSujObj* indicates that if it is realized as an infinitive clause, the object is controlled by the subject: e.g., *chercher du pain* is controlled by *Pierre* in **Pierre** *va* **chercher du pain** (**Peter** *goes* **to get bread**);
- mood of the complementizer phrase: *@SCompInd* indicates that if the subject is realized as a completive clause, its mood is indicative: e.g., *qu'il fait beau* in *Pierre dit* **qu'il fait beau** (*Peter says* **that the weather is good**); on the contrary, *@CompSubj* indicates that if the direct object is realized as a completive clause, its mood is subjunctive: e.g., *qu'il fasse beau* in *Pierre veut* **qu'il fasse beau** (*Peter wants* **that the weather be good**); the following abbreviations are used: *S* for subject, nothing for direct object, *A* for indirect object introduced by the preposition *à*, *De* for indirect object introduced by the preposition *de*; by default, the two modes are possible;
- human or non human noun phrase: *@ObjàNhum* indicates that the indirect object introduced by the preposition *à* can denote a person, or an animal linguistically assimilated to a person when is realized by a noun phrase: e.g., *Vercingetorix s'est rendu à* **l'ennemi** (*Vercingetorix surrendered to* **the enemy**); on the contrary,

@ObjàN-hum indicates that the indirect object introduced by the preposition *à* can be non human: e.g., *Jean s'est rendu à* **mon opinion** (*John finally accepted my opinion*); in this case, *@ObjàNhum* and *@ObjàN-hum* are not in the same entry because the sense of *se rendre* (*to accept/to surrender*) is different; in other cases, for instance, **Max** *va tomber* (**Max** *goes down*) and **Le verre** *va tomber* (**The glass** *will fall*), *@SujNhum* and *@SujN-hum* indicate that the subject can denote a human or a non human for the same entry; if no indication is given for an argument, we can consider both are always possible;

In addition, for verbs, there are other macros such as:

- auxiliary conjugation: *@avoir*: e.g., the verb *achever* in *Max* **a** *achevé de peindre le mur* (*Max has finished painting the wall*) or *@être*: e.g., the verb *s'arrêter* in *Max s'***est** *arrêté de boire* (*Max stopped drinking*);
- pronominal verb: *@pron*: e.g., the verb *arrêter* in the previous example (**s'***arrêter*);
- necessarily negative verb: *@neg*: e.g., the verb *rajeunir* in *Que sa fille ait 20 ans* **ne** *rajeunit* **pas** *Max* (*His daughter being 20 years does***n't** *make Max feel very young*);
- other clitic pronouns necessarily frozen with the verb: *@pseudo-en*: e.g., the verb *baver* in *Max* **en** *bave avec Luc* (*Max has a hard time with Luc*), *@pseudo-y*: e.g., the verb *aller* in *Luc* **y** *va* (*Luc leaves*), *@pseudo-le*: e.g., the verb *disputer* in *La haine* **le** *dispute à la colère* (*Hate and ire are competing*), *@pseudo-la*: e.g., the verb *fermer* in *Fermez-***la** (*Shut up*), *@pseudo-les*: e.g., the verb *aligner* in *Max* **les** *aligne à Luc* (*Max gives the money to Luc*).



Génération du lexique *LGLex*-Le*fff* à l'aide de *LGLex2ilex*

L'utilisation de *LGLex2ilex* pour générer le lexique *LGLex* est :

```
lglex2ilex.pl [options] {lefff_files} < [lglex_file] > [ilex_file]
```

avec `[lglex_file]` le fichier d'entrée, `[ilex_file]` le fichier de sortie, {`lefff_files`} une liste de fichiers *.ilex* du Le*fff* qui sont utilisés pour connaître la classe morphologique de chaque entrée.
Les `[options]` peuvent être :

`-nuc ou - -no_unknown_construction`

pour considérer que les constructions inconnues donnent lieu à la création d'entrées secondaires distinctes au lieu de créer des redistributions inconnues qui ne seront pas déductibles de la construction de base (cf. les deux types de traitement pour les constructions qui ne relèvent pas du cas 5 mais ne sont pas des variantes de la CBME dans 8.1.1).

`-e [file] ou - -examples [file]`

avec `[file]` un fichier contenant des exemples pour chaque entrée de chaque table, dont le format est :

```
<lemma><TAB><table_id><TAB><first_example>(<TAB><other_examples>)
```

ce qui permet de récupérer les exemples de ce fichier plutôt que ceux des tables [1].

Voici un exemple d'utilisation (dans `lefff/otherlex2lefff/`) [2] :

1. Cette option peut être utilisée pour les verbes, puisqu'il existe un index électronique qui recense toutes les entrées en donnant plusieurs exemples (cf. 5.5.2).

2. La conversion des expressions figées étant en cours (cf. 8.1.6), le fichier *expr-lglex.ilex* n'est pas encore utilisable tel quel.

```
perl lglex2ilex.pl -nuc ../v.ilex ../v-phd.ilex
     < $TABLESPATH/lglex/verbes.lglex.txt  > v-lglex.ilex
perl lglex2ilex.pl -nuc ../nom.ilex
     < $TABLESPATH/lglex/noms-predicatifs.lglex.txt  > npred-lglex.ilex
perl lglex2ilex.pl -nuc ../v.ilex ../v-phd.ilex
     < $TABLESPATH/lglex/figees.lglex.txt  > expr-lglex.ilex
```

Les fichiers *v.ilex*, *v-phd.ilex* et *nom.ilex* font partie du Le*fff* mais ne servent ici qu'à attribuer la table de flexion. S'ils ne sont pas présents, les règles par défaut sont appliquées pour les mots dont la flexion n'est pas connue. À savoir, si c'est un mot en *-er*, on lui attribue la catégorie de verbe du premier groupe, etc. et par défaut c'est un mot invariable.

La description manuelle des lemmes qui se fléchissent de manière irrégulière (*.mf*) n'est pas utilisée par *LGLex* mais est nécessaire pour l'étape suivante. Les fichiers associées sont donc créés uniquement pour qu'ils existent (dans `lefff/otherlex2lefff/`) :

```
touch v-lglex.mf
touch npred-lglex.mf
```

Puis, on crée le lexique extensionnel (*.lex*) à partir du lexique intensionnel (*.ilex*) et de la description manuelle des lemmes qui se fléchissent de manière irrégulière (*.mf*) (dans `lefff/`) :

```
make otherlex2lefff/v-lglex.lex
make otherlex2lefff/npred-lglex.lex
```

Les fichiers devant être dans le répertoire /exportbuild/src/lefff/ pour la suite, on crée des liens symboliques une fois pour toute :

```
ln -s otherlex2lefff/v-lglex.lex .
ln -s otherlex2lefff/npred-lglex.lex .
```

Ensuite, on crée le dictionnaire (*.xlfg*) pour FRMG (Le*fff* compilé) contenant toutes les entrées des divers fichiers du lexique extensionnel (*.lex*) provenant du Le*fff* et/ou des tables (dans `lefff-frmg/`) :

```
make lglex
```



Documentation des propriétés des tables de verbes distributionnels

Les Tab. E.1 à Tab. E.43 représentent la documentation complète des 551 propriétés syntaxico-sémantiques des tables de verbes. Pour chaque propriété nous indiquons sa nature (binaire ou lexicale), puis nous donnons sa définition en anglais [1] en l'illustrant d'un exemple.

1. La définition en français figure dans la documentation de la version 3.3 mais n'a pas été incluse ici.

Feature	Description of the feature
(N1+N2) =: le fait Qu P (binary)	A locative object noted N1 or N2 can have an abstract meaning and take the form le fait que P, close to a sentential complement Example : *Max progresse vers* **le fait qu'il faut accepter la situation**
(N1+N2) être V-n (binary)	A locative object noted N1 or N2 can become the subject of a sentence with *être* and a deverbal noun related with V as a 'predicate nominative' Example : *Paul fouille dans le tiroir = Le tiroir est un fouillis*
(N1+N2) être V-n (forme V-n) (lexical)	Form of a deverbal noun related with V, used when a locative object noted N1 or N2 becomes the subject of a sentence with *être* and the deverbal noun as a 'predicate nominative' Example : *fouillis*
(N2+N3) être V-n (binary)	A locative object noted N2 or N3 can become the subject of a sentence with *être* and a deverbal noun related with V as a 'predicate nominative' Example : *On emprisonne Max dans la cave = La cave est une prison*
(N2+N3) être V-n (forme V-n) (lexical)	Form of a deverbal noun related with V, used when a locative object noted N2 or N3 becomes the subject of a sentence with *être* and the deverbal noun as a 'predicate nominative' Example : *prison*
(N2+N3) V N1 (binary)	A locative object noted N2 or N3 can become the subject of a sentence with a direct object N1 ; the sentence is interpreted as a static result Example : *On emprisonne Max dans la cave = La cave emprisonne Max*
(Nég, interro) => subj (binary)	Negation or interrogation in the main clause may change mood of sentential complement to subjunctive Example : *Je crois qu'il* **viendra** / *Je ne crois pas qu'il* **vienne** / *Crois-tu qu'il* **vienne** *?*
[extrap] (binary)	Subject extraposition, i.e. the subject can shift to a complement position while expletive *il* is introduced in subject position Example : **Un malheur** *arrive à Paul* / **Il** *arrive* **un malheur** *à Paul*
[passif de] (binary)	Passivization with agent introduced by the preposition *de* ; all other objects are preserved Example : *Tout le monde a reconnu Marie à sa voix = Marie a été reconnue* **de tout le monde** *à sa voix*
[passif par] (binary)	Passivization with agent introduced by the preposition *par* ; all other objects are preserved. In some cases, this feature is encoded as N1 être Vpp de N0 Prép N2. The direct object may be N2 where there is no N1 Example : *Paul a mangé la soupe dans un bol = La soupe a été mangée* **par Paul** *dans un bol*
<ENGLISH> (lexical)	Hint to a translation in English Example : *Max vole chercher Ida :* **fly**
<ENT>Ppv (lexical)	Clitic forms frozen with the verb V. The information provided by this feature must agree with that provided by the binary features called Ppv =: en figé, etc. Example : *Luc* **n'en** *revient pas de ce culot ; Luc* **se** *réserve pour la nuit*

TABLE E.1: Documentation des propriétés des tables de verbes (1)

Feature	Description of the feature
<ENT>V (lexical)	Form of the verbal entry V. If an adverb is frozen with the verb, it is part of the form. If clitics are frozen with the verb, they are not part of the form. If a negative adverb is obligatorily present in the construction, but has a free lexical value (*pas*, *point*, *nullement*, *aucunement*, *aucun*, *nul*, *personne*, *rien*, *guère*, *jamais*, *plus*, *nulle part*), it is not part of the form. Example : *Ce film dure longtemps :* **durer Advt** *; Luc n'arrête pas d'être dérangé :* **arrêter pas** *; Votre geste ne va pas :* **aller**
<OPT> (lexical)	Sentence exemplifying the sense of the verbal entry V Example : *Max achève de peindre le mur ; Max achève les blessés*
à N1 = Ppv =: le (binary)	An object à N1 can be pronominalized into a clitic pronoun *le* Example : *Paul apprend à lire = Paul l'apprend*
à N1 =: à ce Qu P = là (binary)	An object à N1 containing a sentential complement à ce que P can be pronominalized into *là* Example : *Toute cette histoire aboutit à ce que Luc a perdu = Toute cette histoire aboutit là*
à N1hum = Ppv (binary)	An object à N1 denoting a person can be pronominalized into a clitic pronoun Example : *Ceci correspond à Marie = Ceci lui correspond*
à N1-hum = Ppv (binary)	An object à N1 with a non-human meaning can be pronominalized into a clitic pronoun Example : *Ceci correspond à un refus = Ceci y correspond*
à N2 = là (binary)	An object à N2 can be pronominalized into *là* Example : *Ceci l'a amené à se manifester = Ceci l'a amené là*
à Nhum =: Loc Npc (binary)	A locative object Loc N2 can be replaced by à Nhum, denoting a person, which is interpreted as Loc Nbp, where Nbp denotes the body or a body part of the person Example : *Luc assène un coup à Paul = Luc assène un coup sur le corps de Paul. Luc arrache un cheveu à Paul = Luc arrache un cheveu de la tête de Paul*
à Nhum destination (binary)	A locative object noted Loc N2 or Loc N3, interpreted as a destination, can take the form à Nhum, denoting a person Example : *On a diffusé le match aux Bretons = On a diffusé le match en Bretagne*
à Nhum source (binary)	A locative object Loc N2, interpreted as a source, can take the form à Nhum, denoting a person Example : *Luc arrache un cheveu à Paul*
à quoi N0 V N1 (binary)	When the sentence follows another sentence in a discourse, an object à N2 can take the form *à quoi* and the position in the beginning of the sentence Example : *(...) À quoi Paul a répliqué qu'il n'était pas d'accord*
autre Loc1 (lexical)	Prepositions of a locative object Loc N1 other than those represented in binary features Example : *Les convives farandolent* **autour de** *la table ; Les délinquants se recrutent (***parmi+chez***) les riches*
autre Loc2 (lexical)	Prepositions of a locative object Loc N2 other than those represented in binary features Example : *Max a découché* **de chez** *Léa ; Max s'en vient* **chez** *vous*

Table E.2: Documentation des propriétés des tables de verbes (2)

Feature	Description of the feature
Aux =: avoir (binary)	V *conjugates at compound tenses with auxiliary avoir* Example : *Que Max parte a très bien convenu à Luc*
Aux =: être (binary)	V *conjugates at compound tenses with auxiliary être* Example : *Max est entré livrer un colis*
de combien ? (binary)	An additional complement introduced by the preposition *de*, denoting a quantity or measurement related to the process, and usually with a cardinal numeral, answers a question in *de combien ?* Example : *Luc approche la chaise de 50 cm ; Luc approche la chaise de combien ?*
de N1 = de ce Qu P = de là (binary)	An object de N1 containing a sentential complement de ce que P can be pronominalized into *de là* Example : *La situation dérive de ce que tu as échoué = La situation dérive de là*
de N1 V N0 (binary)	The subject and an object de N1 can swap positions Example : *De ton silence dépend le succès de la mission*
de N2 =: de combien ? (binary)	An object de N2 can denote a quantity or measurement related to the process, contain a cardinal numeral, and answer a question in *de combien ?* Example : *Paul a tapé Luc de 100 euros ; Paul a tapé Luc de combien ?*
de Nhum source (binary)	The locative object de N2, interpreted as a source, can take the form de Nhum, denoting a person Example : *On a éloigné l'assiette de Marie, On a soutiré de l'argent de Luc*
dé-V (binary)	Sentence with verb-centered argument structure, with the same subject, V with prefix *dé-*, the same direct object N1, and sometimes the indirect object N2 with preposition *de* Example : *Luc décabosse la voiture. Le vent défleurit le jardin de tous ces pavots*
en combien ? (binary)	An object en N2 answers a question in *en combien ?* Example : *Luc a coupé le gâteau en combien ? En quatre. *Luc a pulvérisé le verre en combien ? En 79 morceaux*
il être Vpp Prép N1 (E+par N0) (binary)	Passive extraposition with N1, i.e. N0 V W = il être Vpp Prép N1 (E+par N0). The Prép can be empty. The same construction is obtained by applying an anomalous passivization, then extraposition, and possibly an erasement of Prép N2 Example : *Max a contribué au désordre = Il a été contribué au désordre (E+par Max) ; Max a parlé de ce film à Léa = Il a été parlé de ce film (E+par Max) ; Max a conclu ces corollaires d'un théorème = Il a été conclu ces corollaires (E+par Max)*
il être Vpp Prép N2 (E+par N0) (binary)	Passive extraposition with N2 and without N1, i.e. N0 V W = il être Vpp Prép N2 (E+par N0). The same form is obtained by applying an erasement of N1, an anomalous passivization, then extraposition Example : *Le ministre a discuté de quelque chose avec les syndicats = Il a été discuté avec les syndicats (E+par le ministre)*
il V Prép N1 N0 =: il V Prép N1 de V1c-inf W (binary)	The extraposed subject can be an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent with a nominal post-modifier of a nominal object N1 Example : *Il cadre avec les plans de Paul de nous être agréable*

Table E.3: Documentation des propriétés des tables de verbes (3)

Feature	Description of the feature
il V Prép N1 N0 =: il V Prép N1 de V1-inf W (binary)	The extraposed subject can be an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent with an object N1 Example : *Il échoit à Paul de faire la vaisselle*
il V Prép N1 N0 =: il V Prép N1 Qu Pind (binary)	The extraposed subject can be a sentential complement in the indicative mood, and coexist with an object N1 Example : *Il a filtré de la réunion que Luc est candidat*
il V Prép N1 N0 =: il V Prép N1 Qu Psubj (binary)	The extraposed subject can be a sentential complement in the subjunctive mood, and coexist with an object N1 Example : *Il arrive à Max qu'on le prenne pour un autre*
il V Qu P W (binary)	The subject can take the form of a sentential complement and be extraposed, i.e. shift to a complement position while expletive *il* is introduced in subject position ; the sentence has a metaphorical meaning Example : *Il étincelle dans le regard de Luc qu'il est amoureux*
il y avoir V-n entre N0 et N1 (binary)	Sentence with noun-centered argument structure, with *il y avoir*, a deverbal noun related with V, and the preposition *entre* before a conjunction of N0 and N1 Example : *Paul lutte avec Luc = Il y a une lutte entre Paul et Luc*
il y avoir V-n entre N0 et N1 (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a sentence with noun-centered argument structure, with *il y avoir*, the deverbal noun, and the preposition *entre* before a conjunction of N0 and N1 Example : *lutte*
il y avoir V-n entre N1 et N2 (binary)	Sentence with noun-centered argument structure, with *il y avoir*, a deverbal noun related with V, and the preposition *entre* before a conjunction of N1 and N2 Example : *J'ai choqué mon verre contre celui de Max = Il y a eu choc entre mon verre et celui de Max*
il y avoir V-n entre N1 et N2 (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a sentence with noun-centered argument structure, with *il y avoir*, the deverbal noun, and the preposition *entre* before a conjunction of N1 and N2 Example : *choc*
impératif => subj (binary)	The sentential complement is in the subjunctive when the sentence expresses an order Example : *Je crie que Paul vienne (cf. Je crie que Paul vient)*
Loc N1 = ici (binary)	A locative object Loc N1 can be pronominalized into *ici* Example : *Paul entre ici*
Loc N1 = là (binary)	A locative object Loc N1 can be pronominalized into *là* Example : *Paul monte là*
Loc N1 = Ppv (binary)	Clitic pronominalization of a locative object Loc N1 Example : *Cette idée trotte dans sa tête = Cette idée y trotte*
Loc N1 = Ppv =: en (binary)	Clitic pronominalization of a locative object Loc N1 into *en* Example : *Deux poussins ont éclos d'un même oeuf = Deux poussins en ont éclos*
Loc N1 = Ppv =: y (binary)	Clitic pronominalization of a locative object Loc N1 into *y* Example : *On s'écrase dans le métro = On s'y écrase*
Loc N1 =: à N1 (binary)	The preposition of a locative object Loc N1 can be *à* Example : *Paul se précipite à la boulangerie*
Loc N1 =: à N1 destination (binary)	The preposition of a locative object Loc N1, interpreted as a destination, can be *à* Example : *Le sang afflue à sa tête*

TABLE E.4: Documentation des propriétés des tables de verbes (4)

Feature	Description of the feature
Loc N1 =: contre N1 (binary)	The preposition of a locative object Loc N1 can be *contre* Example : *Le volet bat contre le mur*
Loc N1 =: contre N1 destination (binary)	The preposition of a locative object Loc N1, interpreted as a destination, can be *contre* Example : *Max a cartonné contre un pin avec la voiture*
Loc N1 =: dans N1 (binary)	The preposition of a locative object Loc N1 can be *dans* Example : *Paul plonge dans l'eau*
Loc N1 =: dans N1 destination (binary)	The preposition of a locative object Loc N1, interpreted as a destination, can be *dans* Example : *L'auto dérape dans le fossé*
Loc N1 =: de N1 (binary)	The preposition of a locative object Loc N1 can be *de* Example : *Paul rentre de la boulangerie. Paul s'approche de la boulangerie*
Loc N1 =: de N1 source (binary)	The preposition of a locative object Loc N1, interpreted as a source, can be *de* Example : *Paul débouche du garage*
Loc N1 =: de V0-inf W (binary)	A locative object Loc N1 can take the form of an infinitive clause interpreted as a source, taking the form de V0-inf W, with the preposition *de* and an implicit subject coreferent with that of the main verb Example : *Paul revient de chercher du pain (Paul revient de la boulangerie)*
Loc N1 =: Loc N1 source, Loc # de (binary)	The preposition of a locative object Loc N1, interpreted as a source, can be other than *de* Example : *Luc est né dans une famille riche*
Loc N1 =: Loc-source N1, Loc-source # de (binary)	The preposition of a stative locative object Loc N1 can be a source preposition other than *de* Example : *Le fil pend au plafond*
Loc N1 =: par N1 (binary)	The preposition of a locative object Loc N1 can be *par* Example : *Luc a pris le couloir = Luc a pris par le couloir*
Loc N1 =: sur N1 (binary)	The preposition of a locative object Loc N1 can be *sur* Example : *De l'eau dégouline sur le toit*
Loc N1 =: sur N1 destination (binary)	The preposition of a locative object Loc N1, interpreted as a destination, can be *sur* Example : *Ce bas a déteint sur la robe*
Loc N1 =: vers N1 (binary)	The preposition of a locative object Loc N1 can be *vers* Example : *Cette maison regarde vers le nord*
Loc N1 =: vers N1 destination (binary)	The preposition of a locative object Loc N1, interpreted as a destination, can be *vers* Example : *Le bateau dérive vers la côte*
Loc N2 = ici (binary)	A locative object Loc N2 can be pronominalized into *ici* Example : *Il ramène Paul ici faire son numéro*
Loc N2 = là (binary)	A locative object Loc N2 can be pronominalized into *là* Example : *Il conduit Paul là prendre le bus*
Loc N2 = Ppv =: en (binary)	Clitic pronominalization of a locative object Loc N2 into *en* Example : *Max a abstrait son esprit des difficultés présentes = Max en a abstrait son esprit*
Loc N2 = Ppv =: y (binary)	Clitic pronominalization of a locative object Loc N2 into *y* Example : *L'eau coule du robinet dans l'évier = L'eau y coule du robinet ; On accepte les chiens dans ce restaurant = On y accepte les chiens*

Table E.5: Documentation des propriétés des tables de verbes (5)

Feature	Description of the feature
Loc N2 =: à N2 (binary)	The preposition of a locative object Loc N2 can be *à* Example : *L'eau arrive aux genoux de Luc*
Loc N2 =: à N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *à* Example : *Paul greffe un oeil au rosier*
Loc N2 =: à Nhum (binary)	A locative object Loc N2 can take the form à Nhum, denoting a person Example : *Paul envoie Marie au patron demander un formulaire ; Paul encorde Léa au guide*
Loc N2 =: à N-hum (binary)	A locative object Loc N2 can take the form à N-hum, with a non-human meaning Example : *Paul envoie Marie à la boulangerie chercher des croissants*
Loc N2 =: autour de N2 (binary)	The preposition of a locative object Loc N2 can be *autour de* Example : *Paul conserve la main autour du manche*
Loc N2 =: autour de N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *autour de* Example : *Paul enroule le fil autour du stylo*
Loc N2 =: contre N2 (binary)	The preposition of a locative object Loc N2 can be *contre* Example : *Max cale le lit contre le mur*
Loc N2 =: contre N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *contre* Example : *La balle rebondit contre le mur*
Loc N2 =: dans N2 (binary)	The preposition of a locative object Loc N2 can be *dans* Example : *Paul envoie Marie dans le désert chercher du sable*
Loc N2 =: dans N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *dans* Example : *Paul a érigé une statue dans son jardin*
Loc N2 =: de N2 (binary)	The preposition of a locative object Loc N2 can be *de* Example : *Il ramène Paul du Brésil faire un show*
Loc N2 =: de N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *de* Example : *Luc rapproche la bouteille du cendrier*
Loc N2 =: de N2 source (binary)	The preposition of a locative object Loc N2, interpreted as a source, can be *de* Example : *Luc éloigne la bouteille du cendrier*
Loc N2 =: Loc N2 source, Loc # de (binary)	The preposition of a locative object Loc N2, interpreted as a source, can be other than *de* Example : *Luc découpe une photo dans un journal, Luc saisit un livre sur l'étagère, Luc boit à la bouteille*
Loc N2 =: sur lui1-même (binary)	A locative object Loc N2 can take the form *sur lui-même*, where the pronoun *lui* is coreferent to an object N1 Example : *Le froid a recroquevillé la plante sur elle-même, On a lové le filin sur lui-même*
Loc N2 =: sur N2 (binary)	The preposition of a locative object Loc N2 can be *sur* Example : *Luc étale le beurre sur la tartine*
Loc N2 =: sur N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *sur* Example : *Paul débouche sur la place*

TABLE E.6: Documentation des propriétés des tables de verbes (6)

Feature	Description of the feature
Loc N2 =: vers N2 (binary)	The preposition of a locative object Loc N2 can be *vers* with a directional interpretation Example : *L'appartement donne vers le sud. Paul dirige son fusil vers la cible*
Loc N2 =: vers N2 destination (binary)	The preposition of a locative object Loc N2, interpreted as a destination, can be *vers* with a directional interpretation Example : *Paul titube vers la sortie. Luc lance la balle vers la porte*
Loc N3 = Ppv =: y (binary)	Clitic pronominalization of a locative object Loc N3 into *y* Example : *Max conduit Ida de la cuisine dans le salon = Max l'y conduit de la cuisine*
Loc N3 =: à N3 destination (binary)	The preposition of a locative object Loc N3, interpreted as a destination, can be *à* Example : *Max convie Luc à une soirée*
Loc N3 =: contre N3 destination (binary)	The preposition of a locative object Loc N3, interpreted as a destination, can be *contre* Example : *Le car serre le cycliste contre le trottoir ; Luc empile les chaises contre le mur*
Loc N3 =: dans N3 destination (binary)	The preposition of a locative object Loc N3, interpreted as a destination, can be *dans* Example : *Le maître attroupe les enfants dans la cour ; On a interné Alex dans un asile*
Loc N3 =: sur lui1-même (binary)	A locative object Loc N3 can take the form *sur lui-même*, where the pronoun *lui* is coreferent to an object N1 Example : *Paul a basculé l'armoire sur elle-même*
Loc N3 =: sur N3 destination (binary)	The preposition of a locative object Loc N3, interpreted as a destination, can be *sur* Example : *Max dépose Luc sur la place ; Paul concentre les rayons sur la cible*
Loc N3 =: vers N3 (binary)	The preposition of a locative object Loc N3 can be *vers* with a directional interpretation Example : *On draine l'eau vers le canal*
Loc N3 =: vers N3 destination (binary)	The preposition of a locative object Loc N3, interpreted as a destination, can be *vers* with a directional interpretation Example : *Le général a replié ses soldats vers la plaine*
N0 (E+Ppv) V (Prép+Prép-adv) (binary)	Anaphorical erasement or clitic pronominalization of an object N1 with conservation of Prép or modification of Prép into an adverbial form Example : *Luc court après les honneurs = Luc court après ; Luc court après Marie = Luc lui court après ; On a rogné sur le budget = On a rogné dessus*
N0 =: chemin (binary)	The subject can denote a path, in a sentence that describes a stative situation where a person or thing can travel through the path Example : *La jetée avance dans la mer ; La route longe la rivière*
N0 =: idée, (N1+N2) =: esprit (binary)	The subject can denote an idea and the locative object noted N1 or N2 a person's mind ; the sentence has a metaphorical meaning Example : *L'idée de s'en aller filtrait dans son esprit*
N0 =: idée, N1 =: esprit (binary)	The subject can denote an idea and the object N1 a person's mind ; the sentence has a metaphorical meaning Example : *Un flot d'idées ruisselait dans son esprit*

TABLE E.7: Documentation des propriétés des tables de verbes (7)

Feature	Description of the feature
N0 =: le fait Qu P (binary)	The subject can take the form le fait que P, close to a sentential complement, and denote an abstract entity Example : *Le fait que Paul vienne me perturbe*
N0 =: Nhum (binary)	The subject can denote a person, or an animal linguistically assimilated to a person Example : *Luc quitte la ville. L'ennemi quitte la ville*
N0 =: N-hum (binary)	The subject can denote an entity which is not a person, nor an animal linguistically assimilated to a person Example : *La voiture a croisé le camion ; Les fruits débordent sur les étals*
N0 =: N-hum métaphore (binary)	The subject can denote an entity which is not a person, nor an animal linguistically assimilated to a person ; the sentence has a metaphorical meaning (N0 normally denotes obligatorily a person) Example : *Les nuages pérégrinent dans le ciel*
N0 =: Nhum, (N1+N2) =: Nabs (binary)	The subject can denote a person and the locative object noted N1 or N2 an abstract entity ; the sentence has a metaphorical meaning Example : *Paul chemine vers sa pensée*
N0 =: Nhum, N1 =: Nabs (binary)	The subject can denote a person and the locative object noted N1 an abstract entity ; the sentence has a metaphorical meaning Example : *Paul fouille dans sa mémoire*
N0 =: Nhum, N1 =: mot (binary)	The subject can denote a person and the object N1 words ; the sentence has a metaphorical meaning Example : *Paul vomit des insultes ; Luc ingurgite mes paroles*
N0 =: Nnc (binary)	The insertion of the verb into an independently observable sentence does not impose additional constraints on the subject Example : *Il pleut = Il continue de pleuvoir*
N0 =: Nnr (binary)	The subject can denote a person, a concrete object, an abstract entity, or take the form of a sentential subject or an infinitive clause Example : *(Paul+Ce gâteau+L'amour+Que Paul vienne+Faire cela) ennuie Marie*
N0 =: Npl obl (binary)	The subject is obligatorily in the plural or has a collective interpretation Example : *Les missiles convergent sur la cible*
N0 =: Qu P (binary)	The subject can be a sentential subject Example : *Que Marie vienne déplait à Luc*
N0 =: V1c-inf W (binary)	The subject can be an infinitive clause with an implicit subject coreferent with a nominal post-modifier of a nominal object N1 Example : *Se présenter aux élections a germé dans la tête de Paul*
N0 =: V1-inf W (binary)	The subject can be an infinitive clause with an implicit subject coreferent with an object N1 Example : *Faire du yoga calme* **Max** *; Avoir ce diplôme a bénéficié à* **Luc**
N0 =: V2c-inf W (binary)	The subject can be an infinitive clause with an implicit subject coreferent with a nominal post-modifier of a nominal object N2 Example : *Avoir vu ça écarte de l'esprit de* **Paul** *de se présenter*

Table E.8: Documentation des propriétés des tables de verbes (8)

Feature	Description of the feature
N0 =: V2-inf W (binary)	The subject can be an infinitive clause with an implicit subject coreferent with an object N2 Example : *Avoir fait ça prouve à* **Luc** *qu'il est en forme*
N0 =: V3-inf W (binary)	The subject can be an infinitive clause with an implicit subject coreferent with an object N3 Example : (no Example known)
N0 =: Vi-inf W (binary)	The subject can be an infinitive clause with an implicit subject coreferent with the subject of a sentential complement N1 Example : *Être trop gros empêche Luc de passer dans le couloir*
N0 =: V-inf W (binary)	The subject can be an infinitive clause Example : *Faire ce genre de truc pèche contre le bon sens*
N0 avoir V-n pour N1 (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *avoir*, a deverbal noun related with V, and an object pour N1 Example : *Paul a de l'admiration pour Marie*
N0 avoir V-n pour N1 (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a sentence with noun-centered argument structure, with the same subject, the verb *avoir*, the deverbal noun, and an object pour N1 Example : *admiration*
N0 dé-V N1 Loc N3 nv-dest (binary)	Sentence with the same subject, a verb derived from V with a negative prefix, the same object N1 and a locative object N3, interpreted as a new destination ; the locative object N2 of the basic construction is interpreted as a source in this one Example : *On a bobiné le fil sur un crayon ; On a débobiné le fil sur le lit*
N0 dé-V N1 Prép N2 source (binary)	Sentence with the same subject, a verb derived from V with a negative prefix, the same object N1 and the same locative object N2, interpreted as a source and possibly introduced by another preposition Example : *On charge les caisses sur la remorque = On décharge les caisses de la remorque*
N0 donner à N1 (une apparence Vpp+l'apparence de V-n) (binary)	Paraphrase of the form N0 donner une apparence Vpp à N1, or N0 donner une apparence de V-n à N1, where V-n is a deverbal noun related to V Example : *Ce colorant argente les cheveux = Ce colorant donne aux cheveux l'apparence de l'argent = Ce colorant donne aux cheveux une apparence argentée*
N0 donner à N1 (une apparence Vpp+l'apparence de V-n) (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a paraphrase of the form N0 donner une apparence Vpp à N1, or N0 donner une apparence de V-n à N1 Example : *argent*
N0 donner V-n à N1 (binary)	Paraphrase of the form N0 donner V-n à N1, where V-n is a deverbal noun related to V Example : *Paul a armé Luc = Paul a donné une arme à Luc ; Luc a daté le vase du premier siècle = Luc a donné une date au vase*

Table E.9: Documentation des propriétés des tables de verbes (9)

Feature	Description of the feature
N0 donner V-n à N1 (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a paraphrase of the form N0 donner V-n à N1 Example : *armée*
N0 et N1 V (binary)	Sentence with a coordination of N0 and N1 as a subject, V, and no objects Example : *Luc et Marie parlementent*
N0 et N2 V Prép N1 (binary)	Sentence with a coordination of N0 and N2 as a subject, V, and the object Prép1 N1 Example : *Max conspire avec Luc contre le président = Max et Luc conspirent contre le président*
N0 être en V-n avec N1 (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *être*, the preposition *en*, a deverbal noun related with V, and the object avec N1 Example : *Paul correspond avec Luc = Paul est en correspondance avec Luc*
N0 être en V-n avec N1 (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a sentence argument structure, with the same subject, the verb *être*, the preposition *en*, the deverbal noun, and the object avec N1 Example : *correspondance*
N0 être V-(eur+ateur) W (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-eur* or *-ateur*, and sometimes a prepositional object Prép N1 Example : *Ceci flatte Luc = Ceci est flatteur pour Luc ; Ceci modère Luc = Ceci est modérateur*
N0 être V-able W (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-able*, and sometimes a prepositional object Prép N1 Example : *Cette histoire délecte Luc = Cette histoire est délectable (E+pour Luc)*
N0 être V-ant (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, and a deverbal adjective related with V with suffix *-ant* Example : *Paul sourit = Paul est souriant ; La pâte croustille = La pâte est croustillante*
N0 être V-ant Loc N1 (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-ant*, and the locative object Loc N1 Example : *Les escargots abondent dans le jardin = Les escargots sont abondants dans le jardin*
N0 être V-ant pour N1 (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-ant*, and an object pour N1 Example : *Ceci menace Paul de devoir partir = Ceci est menaçant pour Paul*

TABLE E.10: Documentation des propriétés des tables de verbes (10)

Feature	Description of the feature
N0 être V-ant Prép (N1+N2) (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-ant*, and the object noted Prép N1 or Prép N2 Example : *Le soleil décline sur l'horizon = Le soleil est déclinant sur l'horizon*
N0 être V-ant Prép N1 (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-ant*, and the object Prép N1 Example : *Ida dépend de Max = Ida est dépendante de Max*
N0 être V-ant W (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-ant*, and sometimes a prepositional object Prép N1 Example : *Ce truc énerve Luc = Ce truc est énervant pour Luc*
N0 être V-eux W (binary)	Sentence with adjective-centered argument structure, with the same subject, the verb *être*, a deverbal adjective related with V with suffix *-eux*, and sometimes a prepositional object Prép N1 Example : *Cette solution désavantage Paul = Cette solution est désavantageuse (E+pour Paul)*
N0 être V-n (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *être*, and a deverbal noun related with V Example : *Luc a escroqué Max = Luc est un escroc*
N0 être V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a sentence with noun-centered argument structure, with the same subject, the verb *être*, and the deverbal noun Example : *escroc*
N0 être Vpp (binary)	Sentence with the same subject, the verb *être*, the past participle of V ; the sentence has a stative interpretation Example : *L'arbre penche vers le sol = L'arbre est penché ; L'oiseau s'envole de son perchoir = L'oiseau est envolé*
N0 être Vpp W (binary)	Sentence with the same subject, the verb *être*, the past participle of V, and the same complements ; the sentence has a stative interpretation Example : *Paul s'entête à ce que tout soit en ordre = Paul est entêté à ce que tout soit en ordre ; Paul se décide à travailler = Paul est décidé à travailler ; Paul s'empresse de partir = Paul est empressé de partir*
N0 faire le V-n de N1 (binary)	Sentence with noun-centered argument structure, with the same subject, the verb faire, the determiner *le*, a deverbal noun related with V, and an object de N1 Example : *Paul pastiche le style de Luc = Paul fait le pastiche du style de Luc*
N0 faire le V-n de N1 (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a sentence with noun-centered argument structure, with the same subject, the verb *faire*, the determiner *le*, the deverbal noun, and an object de N1 Example : *pastiche*
N0 faire V-n Prép N1 (binary)	Paraphrase of the form N0 faire V-n Prép N1, where V-n is a deverbal noun related to V and Prép a preposition Example : *Paul assaisonne la salade = Paul fait l'assaisonnement de la salade ; Paul barre son chèque = Paul fait une barre sur son chèque ; La fatigue cerne ses yeux = La fatigue fait des cernes à ses yeux*

TABLE E.11: Documentation des propriétés des tables de verbes (11)

Feature	Description of the feature
N0 faire le V-n Prép N1 (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphrase of the form N0 faire V-n Prép N1, where Prép is a preposition Example : *assaisonnement*
N0 lancer V-n Loc N1 (binary)	Paraphrase of the form N0 lancer V-n Loc N1, where V-n is a deverbal noun related to V and Loc a locative preposition Example : *On a torpillé le bateau* = *On a lancé une torpille sur le bateau*
N0 lancer V-n Loc N1 (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphrase of the form N0 lancer V-n Loc N1, where Loc is a locative preposition Example : *torpille*
N0 lieu de destination de N1 (binary)	The subject is interpreted as the destination of the referent of N1. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Max tire la corde à lui ; Paul se tape une bière ; Paul a réceptionné le colis*
N0 lieu de destination de N2 (binary)	The subject is interpreted as the destination of the referent of N2. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Luc dépouille Léa de ses biens*
N0 lieu source de N1 (binary)	The subject is interpreted as the source of the referent of N1. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Luc catapulte la balle dans les airs ; Max congédie Paul*
N0 lieu statique de N1 (binary)	The subject is interpreted as a place where the referent of an object N1 is before, during and after the process. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Luc arbore une cravate*
N0 lui V Loc (N1pc+N2pc) (binary)	Pronominalization of a noun which denotes a person and is a post-modifier of a locative object, denoting a body part and noted Loc N1 or Loc N2 ; the pronoun is a dative clitic Example : *La balle a ricoché sur la poitrine* **de Luc** = *La balle* **lui** *a ricoché sur la poitrine*
N0 lui V Loc N1pc (binary)	Pronominalization of a noun which denotes a person and is a post-modifier of a locative object Loc N1 denoting a body part ; the pronoun is a dative clitic Example : *De l'eau dégouline sur la tête* **de Luc** = *De l'eau* **lui** *dégouline sur la tête*
N0 lui V N1 Loc N3pc (binary)	Pronominalization of a noun which denotes a person and is a post-modifier of a locative object Loc N3 denoting a body part and interpreted as a destination ; the pronoun is a dative clitic Example : *Paul empile des assiettes sur la tête* **de Luc** = *Paul* **lui** *empile des assiettes sur la tête*

TABLE E.12: Documentation des propriétés des tables de verbes (12)

Feature	Description of the feature
N0 lui V N1pc W (binary)	Pronominalization of a noun which denotes a person and is a post-modifier of an object N1 denoting a body part ; the pronoun is a dative clitic Example : *Le froid a rougi le nez* **de Paul** = *Le froid* **lui** *a rougi le nez ; L'eau ballonne l'estomac* **de Luc** = *L'eau* **lui** *ballonne l'estomac ; L'obus a brisé le bras* **de Luc** *en trois* = *L'obus* **lui** *a brisé le bras en trois*
N0 lui V Prép N1pc (binary)	Pronominalization of a noun which denotes a person and is a post-modifier of an object Prép N1 denoting a body part ; the pronoun is a dative clitic Example : *Luc lorgne sur les seins* **de Marie** = *Luc* **lui** *lorgne sur les seins*
N0 mettre V-n Loc N1 (binary)	Paraphrase of the form N0 mettre V-n Loc N1, where V-n is a deverbal noun related to V and Loc a locative preposition Example : *Paul borne le pré* = *Paul met des bornes autour du pré*
N0 mettre V-n Loc N1 (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphrase of the form N0 mettre V-n Loc N1, where Loc is a locative preposition Example : *borne*
N0 mouvement (binary)	The subject is interpreted as a moving object or place Example : *La balle a frôlé la cible*
N0 porter V-n à N1 (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *porter*, a deverbal noun related with V, and an object à N1 Example : *Paul porte une grande admiration à son chef*
N0 porter V-n à N1 (forme V-n) (lexical)	Form of a deverbal noun related with V, used in a sentence with noun-centered argument structure, with the same subject, the verb *porter*, the deverbal noun, and an object à N1 Example : *admiration*
N0 pousser V-n, Sfx = E (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *pousser*, and a deverbal noun related with V with an empty suffix Example : *Luc crie* = *Luc pousse un cri*
N0 pousser V-n, Sfx = ment (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *pousser*, and a deverbal noun related with V with the suffix -*ment* Example : *Luc grogne* = *Luc pousse un grognement*
N0 rendre N1 (E+plus) V-adj (binary)	Sentence with the same subject, the verb *rendre*, the same object, sometines a comparative adverb, and a deverbal adjective related with V Example : *L'été a embelli le jardin* = *L'été a rendu le jardin (E+plus) beau ; Ce produit a féminisé Max* = *Ce produit a rendu Max (E+plus) féminin*
N0 rendre N1 plus V-adj (binary)	Sentence with the same subject, the verb *rendre*, the same object, a comparative adverb, and a deverbal adjective related with V Example : *Le vent a abaissé la température* = *Le vent a rendu la température plus basse*
N0 rendre N1 plus V-adj (forme V-n) (lexical)	Form of a deverbal adjective related with V, used in a sentence with the same subject, the verb *rendre*, the same object, a comparative adverb, and the deverbal adjective Example : *bas*

Table E.13: Documentation des propriétés des tables de verbes (13)

Feature	Description of the feature
N0 rendre N1 V-adj (binary)	Sentence with the same subject, the verb *rendre*, the same object, and a deverbal adjective related with V Example : *Ce produit active la fermentation = Ce produit a rendu la fermentation active*
N0 rendre N1 V-adj (forme V-n) (lexical)	Form of a deverbal adjective related with V, used in a sentence with the same subject, the verb *rendre*, the same object, and the deverbal adjective Example : *actif*
N0 V (binary)	Sentence with no objects, possibly by erasement of all the objects Example : *Paul dort; Paul flirte avec Marie = Paul flirte; Paul Luc a inventé que Marie était mariée / *Luc a inventé; Paul hésite à se prononcer = Paul hésite*
N0 V (Adj) (Qu P) (binary)	Raising of two parts of a sentential complement : (i) Adj ; (ii) an embedded sentential subject Example : *Je juge qu'il est souhaitable que Léa vienne = Je juge souhaitable que Léa vienne*
N0 V (être Adj) (Qu P) (binary)	Raising of two parts of a sentential complement : (i) être Adj ; (ii) an embedded sentential subject Example : *Je juge qu'il est souhaitable que Léa vienne = Je juge être souhaitable que Léa vienne ; Max lui dit être possible que Luc arrive demain*
N0 V (N1 de N1c) = N0 V (N1c) (Prép N1) (binary)	A nominal direct object of the form N1 de N1c, where N1c is a nominal post-modifier, can be replaced by N1c and a prepositional modifier Prép N1 called echo modifier (restructuration) Example : *Luc stimule la curiosité de Marie = Luc stimule Marie dans sa curiosité*
N0 V (N2+N3) (E+de N1) (binary)	Sentence where a locative object N2 or N3 is direct, by erasement of its preposition, and an object N1 is erased or occurs with the preposition *de* Example : *On a exproprié l'immeuble de tous ces gens = On a exproprié tous ces gens (E+de l'immeuble) ; On a infiltré 3 agents dans le réseau = On a infiltré le réseau (E+de 3 agents)*
N0 V à N1 (binary)	Sentence with an object à N1, possibly by erasement of other objects Example : *Max ment à Luc ; Ça a abouti pour Paul à un échec = Ça a abouti à un échec*
N0 V à N1 Prép N2 (binary)	Sentence with a sentential complement with the preposition *à* and another object noted Prép N2 Example : *Travailler plus revient pour Luc à ce qu'il gagne plus*
N0 V à N2 (binary)	Sentence with an object à N2, by erasement of an object N1 Example : *Luc a répondu à Paul qu'il pleut = Luc a répondu à Paul ; Ça contraint Luc à des sacrifices = Ça contraint à des sacrifices*
N0 V à N2 : 'P' (binary)	The sentential complement N1 can take the form of direct speech Example : *Paul crie à Marie qu'elle est idiote = Paul crie à Marie : "Tu es idiote"*
N0 V Adv (binary)	The object N1 can take the form of an adverb Example : *Le film dure 3 heures = Le film dure longtemps. Le sac pèse 10 kg = le sac pèse lourd*
N0 V après Nhum (binary)	Sentence with the same subject, the verb V, and an object après Nhum denoting a person Example : *Max gueule après sa femme*

TABLE E.14: Documentation des propriétés des tables de verbes (14)

Feature	Description of the feature
N0 V contre Nhum (binary)	Sentence with the same subject, the verb V, and an object contre Nhum denoting a person Example : *Paul râle contre les syndicats*
N0 V dans N1 (binary)	Sentence with the same subject, the verb V, and a nominal object with the preposition *dans* Example : *Paul persiste dans ses convictions*
N0 V de N0pc (binary)	Sentence with the same subject denoting a concrete thing or a person, the verb V, and an object with the preposition de denoting a part of the concrete thing or of the body of the person Example : *Paul souffre du bras droit ; Le vélo dérape de la roue avant*
N0 V de N0pc W (binary)	With a subject denoting a concrete thing or a person, additional object with the preposition *de* denoting a part of the concrete thing or of the body of the person Example : *La voiture enfonce dans la boue de la roue avant gauche ; Luc presse du doigt sur le bouton*
N0 V de N1 (binary)	Sentence with an object de N1, possibly by erasement of other objects Example : *Max profite du soleil ; Paul parle du soleil à Marie = Paul parle du soleil*
N0 V de N1 Prép N2 (binary)	Sentence with a sentential complement with the preposition *de* and another object noted Prép N2 Example : *Luc parle à Marie de ce qu'il pourrait partir en vacances*
N0 V Loc N1 (binary)	Sentence with a locative object Loc N1, possibly by erasement of other objects or by insertion of a locative preposition Example : *Les étoiles étincellent dans le ciel ; Le bateau aborde au quai = Le bateau aborde le quai ; Max grimpe à l'échelle = Max grimpe l'échelle*
N0 V Loc N1 source (binary)	Sentence with a locative object interpreted as a source, possibly by erasement of other objects Example : *Le train a déraillé de la voie ; La fumée sort de la cheminée*
N0 V Loc N1 destination (binary)	Sentence with a locative object interpreted as a destination Example : *Max s'est effondré dans un fauteuil*
N0 V Loc N1 Loc N2 (binary)	Stative sentence with two locative objects, the first with a source preposition, and the second with a destination preposition Example : *L'escalier descend de la cuisine à la cave*
N0 V Loc N1 source Loc N2 destination (binary)	Sentence with two locative objects, the first interpreted as a source, and the second as a destination Example : *Luc descend de la cuisine à la cave*
N0 V Loc N1 V0-inf W (binary)	Sentence with a nominal locative object and an infinitive clause with an implicit subject coreferent with that of the main verb Example : *Paul entre dans le bureau demander du feu ; Paul grimpe à l'échelle tailler le lierre*
N0 V Loc N2 (binary)	Sentence with a locative object Loc N2, by erasement of other objects Example : *Max descend Luc à la gare = Max descend à la gare ; La ville s'étend de la rivière jusqu'à la montagne = La ville s'étend jusqu'à la montagne*
N0 V Loc N2 destination (binary)	Sentence with a locative object Loc N2 interpreted as a destination place, by erasement of other objects Example : *L'eau cascade des rochers sur la vallée = L'eau cascade sur la vallée*

Table E.15: Documentation des propriétés des tables de verbes (15)

Feature	Description of the feature
N0 V N0pc W (binary)	The subject can denote a person, and the direct object, with the determiner *le*, can denote a body part of the person Example : *Max donne le bras à Léa ; Max approche la main du feu ; Max bouge les oreilles*
N0 V N1 (binary)	Sentence with a direct object N1, possibly by erasement of other objects Example : *Luc énerve Ida ; Luc amène Marie au parc = Luc amène Marie ; Luc mène Marie au parc / *Luc mène Marie*
N0 V N1 + 2 (binary)	Sentence with a direct object metonymically interpreted as both objects Example : *Léa a boutonné un pan de sa robe avec l'autre = Léa a boutonné sa robe*
N0 V N1 à ce Qu P (binary)	An object à N2 can be a sentential complement Example : *Paul voit que Luc est passé à ce qu'il y a une bouteille vide*
N0 V N1 à N2 (binary)	Sentence with a direct object and an object à N2 Example : *Luc a cédé ce livre à Marie*
N0 V N1 à N2 de Nhum (binary)	An additional object with de as its preposition denotes a person obligatorily also referred to by an element of the object N1 (projection constraint in the sense of Gross (1975)) Example : *Paul dit à Marie de Luc qu'il est génial, cf. Voici Luc dont Paul dit à Marie qu'il est génial*
N0 V N1 à N2 Dnum Nmonnaie (binary)	An additional direct complement, which takes the form of a cardinal numeral and a currency name, denotes a sum of money Example : *Paul a facturé ce livre 100 francs à Luc*
N0 V N1 à N2 pour Dnum Nmonnaie (binary)	An additional object, with the preposition *pour*, a cardinal numeral and a currency name, denotes a sum of money Example : *Luc a cédé ce livre à Marie pour 60 francs*
N0 V N1 comme N2 (binary)	A direct object N2 can be introduced by comme Example : *On a élu Paul président = On a élu Paul comme président*
N0 V N1 contre N2hum (binary)	An additional object of the form contre N2 denotes a person Example : *Cette décision révolte Luc contre le gouvernement*
N0 V N1 de coup de V-n (binary)	An additional modifier of the form de coup de V-n, denoting a blow, contains a deverbal noun related to the verb V and denotes an instrument Example : *Luc a fouetté Léa de coups de fouet ; Léa a poignardé Luc d'un coup de poignard*
N0 V N1 de coup de V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in an additional modifier of the form de coup de V-n, denoting a blow, that contains the deverbal noun and denotes an instrument Example : *fouet*
N0 V N1 de N2 (binary)	Sentence with a direct object and an object de N2 Example : *Léa accuse Max de viol ; Paul a alourdi son sac d'un gros livre*
N0 V N1 de N2 source (binary)	Sentence with a direct object and a locative object de N2 interpreted as a source, possibly by erasement of other objects Example : *Paul a collecté les informations de ce thésaurus ; Paul perçoit ce salaire d'une PME ; Paul déplace le pion de la case 5*

Table E.16: Documentation des propriétés des tables de verbes (16)

Feature	Description of the feature
N0 V N1 de N2hum (binary)	An additional object with de as its preposition denotes a person obligatorily also referred to by an element of the object N1 (projection constraint) Example : *Paul sait de Marie qu'elle est partie, cf. Marie dont Paul sait qu'elle est partie ; Paul aime de Marie qu'elle s'énerve, cf. Marie dont Paul aime qu'elle s'énerve*
N0 V N1 de N2 matériau (binary)	Sentence with a direct object and an object de N2 denoting material for the process Example : *Paul a bâti cette cabane seulement de planches usagées*
N0 V N1 de N2 = N0 enlever N2 de N1 (binary)	Sentence of the form N0 V N1 de N2 paraphrased by N0 enlever N2 de N1 Example : *Ce procédé épure les eaux de leurs germes = Ce procédé enlève leurs germes des eaux ; Luc a purgé ce radiateur d'un peu d'air = Luc a enlevé un peu d'air de ce radiateur*
N0 V N1 de N2 = N0 mettre N2 Loc N1 (binary)	Sentence of the form N0 V N1 de N2 paraphrased by N0 mettre N2 Loc N1 Example : *Paul a alourdi son sac d'un gros livre = Pierre a mis un gros livre dans son sac ; Luc orne la table de fleurs = Luc met des fleurs sur la table*
N0 V N1 de N3 attache (binary)	Sentence with a direct object and a complement de N3 attache, introduced by the preposition *de*, denoting a tying system and interpreted as an instrument Example : *Luc a boulonné les planches de 4 boulons*
N0 V N1 de N-hum (binary)	Sentence with a direct object and acomplement introduced by the preposition *de* and denoting a non-human entity Example : *Luc appelle son fils d'un surnom*
N0 V N1 de V-n (binary)	An additional complement of the form de V-n contains a deverbal noun related to the verb V Example : *Paul a giflé Léa de 3 baffes ; Luc surnomme sa fille d'un surnom ridicule*
N0 V N1 de V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in an additional complement of the form de V-n Example : *baffe*
N0 V N1 destination (binary)	A locative object N1 interpreted as a destination can become a direct object, by erasement of the preposition introducing N1 Example : *Max farfouille dans le tiroir = Max farfouille le tiroir*
N0 V N1 en N2 (binary)	Sentence with a direct object and an object en N2 Example : *Paul a empilé les dossiers en un gros tas ; Luc a transformé ce taudis en un palais ; Luc a scié la planche en deux bouts*
N0 V N1 entre N2pl obl (binary)	Sentence with a direct object and an object entre N2, where N2 is in the plural or a conjunction of several noun phrases Example : *Paul répartit les dossiers entre Luc et Max*
N0 V N1 et N2 (binary)	Two objects N1 and N2 can be coordinated without preposition Example : *Luc lace le côté droit et le côté gauche ; Le maire marie Max et Léa*
N0 V N1 Loc N1pc W (binary)	An object N1 can denote a person or a concrete object and coexist with an additional locative object Loc N1pc denoting a body part of the person or a part of the concrete object Example : *Ida masse Luc à la jambe ; Le chirurgien quadrille le patient en 9 régions sur la poitrine*

Table E.17: Documentation des propriétés des tables de verbes (17)

Feature	Description of the feature
N0 V N1 Loc N2 (binary)	Sentence with a direct object and a locative object Loc N2 Example : *On a érigé une statue derrière le poteau*
N0 V N1 Loc N2 destination (binary)	Sentence with a direct object and a locative object interpreted as a destination place Example : *Luc glisse la lettre dans la fente*
N0 V N1 Loc N2 source (binary)	Sentence with a direct object and a locative object interpreted as a source place Example : *Luc raye deux noms dans la liste*
N0 V N1 Loc N2 source Loc N3 destination (binary)	Sentence with a direct object and two locative objects, the first being interpreted as a source place and the second as a destination place Example : *Luc chasse la bille du trou dans la rainure ; Luc déménage les meubles du salon dans la soupente*
N0 V N1 Loc N2 source, Loc # de (binary)	Sentence with a direct object and an object interpreted as a source place, but introduced by a preposition other than *de* Example : *Paul recense deux célibataires dans la liste ; Paul grappille des informations dans les journaux*
N0 V N1 Loc N2 V1-inf W (binary)	Sentence with a direct object N1, a nominal locative object and an infinitive clause with an implicit subject coreferent with N1 Example : *Paul envoie Marie dans le bureau demander du feu*
N0 V N1 Loc N2 matériau, Loc # de (binary)	Sentence with a direct object and a locative object denoting material for the process, but introduced by a locative preposition other than *de* Example : *Paul a sculpté un ange dans le marbre*
N0 V N1 Loc N2pc, destination de N2c (binary)	A locative object Loc N2, interpreted as a destination, can have a nominal post-modifier de N2c which denotes a person while N2 denotes a body part of the person Example : *Paul a tatoué une fleur sur l'épaule de Marie*
N0 V N1 Loc N2texte destination (binary)	The object N2 can denote a text and be interpreted as a destination ; the sentence has a metaphorical meaning Example : *Paul a bâti une série d'hypothèses dans son bouquin*
N0 V N1 Loc N3 destination (binary)	Sentence with a direct object and a locative object Loc N3 interpreted as a destination, by erasement of the locative object interpreted as a source Example : *Max a accompagné Luc chez lui*
N0 V N1 Loc N-hum (binary)	Sentence with a direct object and a locative object Loc N, which has a locative preposition, and is interpreted as a source place or as a destination place Example : *Luc flanque un coup sur la table (cf. Luc flanque un coup à Paul) ; Luc barbote 100 F dans la caisse (cf. Luc barbote 100 F à Paul) ; On a nommé Paul à ce poste*
N0 V N1 N2 (binary)	Sentence with two direct objects Example : *Luc appelle son fils Paul*
N0 V N1 Prép N2 (binary)	Sentence with a direct object N1 and a prepositional object Prép N2, possibly by erasement of a preposition introducing N1 Example : *Paul tient cela de Luc*
N0 V N1 Prép N2 contre Nhum (binary)	An additional complement with the preposition *contre* denotes a person Example : *Luc s'accorde avec Marie contre Luc à forger des preuves*

Table E.18: Documentation des propriétés des tables de verbes (18)

Feature	Description of the feature
N0 V N1 Prép N2 Prép N3 (binary)	Sentence with a direct object N1 and two prepositional objects Example : *Max traduit ce texte de l'anglais au turc*
N0 V N1 source (binary)	A locative object N1 interpreted as a source can become the unique object and a direct object, by erasement of other objects and of the preposition introducing N1 Example : *Max a déserté de son régiment = Max a déserté son régiment*
N0 V N1 sur N1pc (binary)	Sentence with a direct object denoting a person or a concrete object, and a locative object denoting a body part of the person or a part of the concrete object, and introduced by the preposition *sur*. The locative object indicates which part of the body or concrete object is concerned by the locative relation. The prepositional object present in the basic construction is absent from this one Example : *Max allonge Marie sur le dos ; Max dispose les livres sur la tranche*
N0 V N1 V0-inf W (binary)	Sentence with a nominal direct locative object and a non-prepositional infinitive clause with an implicit subject coreferent with that of the main verb Example : *Paul traverse la rue acheter du pain*
N0 V N1 V1-inf W (binary)	Sentence with a nominal direct object and a non-prepositional infinitive clause with an implicit subject coreferent with the nominal object Example : *Paul envoie Luc acheter du pain*
N0 V N1-hum (binary)	Sentence with the same subject, the verb V, and a direct nominal object with a non-human meaning Example : *Paul commence le tri des vis ; Paul commence les vis ; Paul choisit entre faire du ski et faire du saut = Paul choisit le ski ou le saut ; Paul choisit entre bombarder le pont et bombarder l'usine = Paul choisit le pont ou l'usine*
N0 V N1abs Loc (N2abs+N3abs) (binary)	Sentence with an object N1 denoting an abstract entity, and an object Loc N2 or Loc N3 introduced by a locative preposition and denoting an abstract entity ; the sentence has a metaphorical meaning Example : *Luc collecte des exemples dans ses souvenirs ; Luc relègue ses rancunes dans les mauvais souvenirs*
N0 V N1abs Loc N2abs (binary)	Sentence with an object N1 denoting an abstract entity, and an object Loc N2 introduced by a locative preposition and denoting an abstract entity ; the sentence has a metaphorical meaning Example : *Paul a adapté la théorie aux nouveaux faits*
N0 V N1c (binary)	A nominal direct object N1 with a nominal post-modifier de N1c can be replaced by N1c alone (metonymic reduction) Example : *Paul contrecarre les projets de Marie = Paul contrecarre Marie ; cf. Paul prévient tous les désirs de Marie/Paul prévient Marie*
N0 V N1c à N1pc (binary)	A nominal direct object of the form N1pc de N1c, where N1c denotes a person or a concrete object and N1pc denotes a body part of the person or a part of the concrete object, can be replaced by N1c and a locative object à N1pc (restructuration) Example : *La flèche a atteint le bras de Luc = La flèche a atteint Luc au bras*

Table E.19: Documentation des propriétés des tables de verbes (19)

Feature	Description of the feature
N0 V N1c par N1pc (binary)	A nominal direct object of the form N1pc de N1c, where N1c denotes a person or a concrete object and N1pc denotes a body part of the person or a part of the concrete object, can be replaced by N1c and a locative object par N1pc (restructuration) Example : *Luc a pris la main de Marie = Luc a pris Marie par la main*
N0 V N1c sur N1pc (binary)	A nominal direct object of the form N1pc de N1c, where N1c denotes a person or a concrete object and N1pc denotes a body part of the person or a part of the concrete object, can be replaced by N1c and a locative object sur N1pc (restructuration) Example : *Luc a embrassé la joue de Marie = Luc a embrassé Marie sur la joue*
N0 V N1hum (binary)	Sentence with the same subject, the verb V, and a direct nominal object denoting a person Example : *Paul choisit entre viser l'homme et viser l'arme = Paul choisit l'homme ou l'arme*
N0 V N1hum de N2abs (binary)	Sentence with an object N1 denoting a person, and a locative object de N2 denoting an abstract entity ; the sentence has a metaphorical meaning Example : *Paul a sorti Luc de ses problèmes ; L'aube nous ensoleilla d'espoir*
N0 V N1hum Loc (N2abs+N3abs) (binary)	Sentence with an object N1 denoting a person, and a locative object Loc N2 or Loc N3 introduced by a preposition and denoting an abstract entity ; the sentence has a metaphorical meaning Example : *Ces épreuves rassemblent les habitants dans un destin commun*
N0 V N1hum Loc N2abs (binary)	Sentence with an object N1 denoting a person, and a locative object Loc N2 introduced by a preposition and denoting an abstract entity ; the sentence has a metaphorical meaning Example : *Cette action a gravé Paul dans les mémoires ; Son éducation a enlisé Léa dans ses complexes*
N0 V N1pc de N1c (binary)	A direct object N1 can take the form N1pc de N1c, where N1c denotes a person and N1pc denotes a body part of the person Example : *Paul dessine le bras de Léa*
N0 V N2 (binary)	An object N2 can become the unique object and a direct object, by erasement of other objects and possibly of the preposition introducing N2 Example : *Luc excuse Paul de ces maladresses = Luc excuse ces maladresses ; On a nommé Max ambassadeur = On a nommé un ambassadeur ; Luc rallie des mercenaires en une armée = Luc rallie une armée*
N0 V N2 (E+de N1 source) (binary)	Sentence where an object N2 is direct, by erasement of its preposition, and an object N1 denoting a source is erased or occurs with the preposition *de* Example : *J'ai débarrassé le grenier de toutes ces caisses = J'ai débarrassé toutes ces caisses du grenier*

Table E.20: Documentation des propriétés des tables de verbes (20)

Feature	Description of the feature
N0 V N2 (E+de N1) (binary)	Sentence where a locative object N2 is direct, by erasement of its preposition, and an object N1 is erased or occurs with the preposition *de* Example : *Luc livre du vin à Paul = Luc livre Paul (E+de vin) ; Luc emmanche un manche sur la pelle = Luc emmanche la pelle (E+d'un manche) ; Luc asperge du vin sur le mur = Luc asperge le mur (E+de vin)*
N0 V N2 (E+Loc N1 destination) (binary)	Sentence where an object N2 is direct, by erasement of its preposition, and an object N1 denoting a place of destination is erased or occurs with a locative preposition Example : *Luc badigeonne le mur de noir = Luc badigeonne du noir sur le mur ; Luc charge le camion de caisses = Luc charge des caisses (E+sur le camion)*
N0 V N2 (E+Loc N1) (binary)	Sentence where an object N2 is direct, by erasement of its preposition, and an object N1 denoting a place is erased or occurs with a locative preposition Example : *Luc coupe le pain en dix tranches = Luc coupe dix tranches (E+dans le pain) ; Max cultive son champ en blé = Max cultive du blé (E+dans son champ)*
N0 V N2 =: N0 V Qu Psubj (binary)	Sentence with a unique object N2, taking the form of a direct sentential complement in the subjunctive mood with as subject the object N1 of the basic construction Example : *Paul soupçonne Ida de boire = Paul soupçonne qu'Ida boit ; Paul excuse Luc d'avoir fait ça = Paul excuse que Luc ait fait ça*
N0 V N2 destination (binary)	A locative object N2 interpreted as a destination can become the unique object and a direct object, by erasement of other objects and of the preposition introducing N2 Example : *Luc a bâti des immeubles sur toute la colline = Luc a bâti toute la colline ; Luc crayonne des dessins sur sa feuille = Luc crayonne sa feuille ; La colonne piétine des pieds dans la boue = La colonne piétine la boue*
N0 V N2 matériau (binary)	An object N2 denoting material for the process can occur as a direct object by erasement of its preposition and of an object N1 Example : *Paul tricote un pull dans de la laine bleue = Paul tricote de la laine bleue*
N0 V N2 source de N1 (binary)	A locative object N2 interpreted as a source can become a direct object, by erasement of its preposition, and coexist with an object de N1 (argument transposition) Example : *On a drainé toute cette eau de notre champ dans une fosse = On a drainé notre champ de toute cette eau*
N0 V Nhum sur ce point (binary)	Sentence with a direct object N1 denoting a person, and an additional complement of the form *sur ce point* Example : *Paul a consulté Luc sur cette question*
N0 V N résultat (binary)	An object interpreted as a result can appear as a unique object which is a direct object ; it can also appear as an additional object with the preposition *en* Example : *Paul tisse un tapis ; Paul tisse les fils bleus et les fils rouges en un tapis*

Table E.21: Documentation des propriétés des tables de verbes (21)

Feature	Description of the feature
N0 V Prép N0pc (binary)	A prepositional object Prép N1 can denote a body part of the person denoted by the subject, or a part of the concrete object denoted by the subject Example : *Paul glisse sur le ventre ; La poutre repose sur le côté*
N0 V Prép N1 (binary)	Sentence with a unique object Prép N1, possibly by erasement of other objects ; the preposition can be empty Example : *Ceci servira à Paul (E+à écrire son article). Luc a retenu Luc (E+de faire une bêtise)*
N0 V Prép N1 Prép N2 (binary)	Sentence with two objects Prép N1 and Prép N2, possibly by erasement of other objects ; the preposition introducing N1 can be empty Example : *Ce bâton sert à Paul de matraque (E+pour se défendre)*
N0 V Prép N1 Prép N2 Prép N3 (binary)	Sentence with three objects Prép N1, Prép N2 and Prép N3 ; the prepositions can be empty Example : *Paul désigne Luc à Marie pour faire la corvée*
N0 V Prép N1 Prép N3 (binary)	Sentence with two objects Prép N1 and Prép N3, by erasement of other objects ; the prepositions can be empty Example : *Paul désigne Luc (E+à Marie) pour faire la corvée*
N0 V Prép N1-hum (binary)	Sentence with the same subject, the verb V, and a nominal object Prép N1 with a non-human meaning Example : *Paul commence par faire le tri des vis = Paul commence par le tri des vis ; Paul commence par faire le tri des vis = Paul commence par les vis*
N0 V Prép N1hum (binary)	Sentence with the same subject, the verb V, and a nominal object Prép N1 denoting a person Example : *Luc commence par répondre à Marie = Luc commence par Marie ; Luc passe pour être un as = Luc passe pour un as*
N0 V Prép N2 (binary)	Sentence with an object Prép N2, by erasement of other objects Example : *Ce traitement immunise (E+les enfants) contre la grippe*
N0 V Prép N2 Prép N3 (binary)	Sentence with two objects Prép N2 and Prép N3, by erasement of other objects ; the preposition introducing N3 can be empty Example : *Paul pose (E+à Luc) comme condition que Léa parte*
N0 V Prép N3 (binary)	Sentence with an object Prép N3, by erasement of other objects Example : *Ce parapluie sert (E+à Marie) (E+d'arme) pour se défendre*
N0 V Prép V0-inf W (binary)	Insertion of the verb V into a sentence N0 V W, producing a sentence with the same subject, the verb V, and an infinitive clause taking the form Prép V0-inf W, with an implicit subject coreferent with N0 ; the preposition can be empty Example : *Paul prend une douche = Paul commence par prendre une douche ; Paul prend un bain = Paul ose prendre un bain*
N0 V V0-inf W (binary)	Sentence with a non-prepositional infinitive clause which has an implicit subject coreferent with that of the main verb ; possibly by erasement of other objects Example : *Paul file acheter du pain ; Paul traverse la rue acheter du pain = Paul traverse acheter du pain*
N0 V vers N (binary)	Additional complement of the form vers N, interpreted as a direction to a destination ; the sentence has no directional meaning without this complement Example : *Paul titube vers la sortie/ Paul titube (cf. L'avion pique vers le sol/L'avion pique)*

Table E.22: Documentation des propriétés des tables de verbes (22)

Feature	Description of the feature
N0abs V N1hum (binary)	Sentence with a subject denoting an abstract entity and an object N1 denoting a person ; the sentence has a metaphorical meaning Example : *Les soucis bouffent Léa*
N0hum V Loc Nabs (binary)	Sentence with a subject denoting a person and a locative complement denoting an abstract entity ; the sentence has a metaphorical meaning Example : *Paul voyage dans des rêves impossibles*
N0hum V W sur ce point (binary)	Sentence with a subject denoting a person, and an additional complement of the form *sur ce point* Example : *Paul à menti à Léa sur ce point ; Paul s'emporte contre Luc sur cette question*
N0idée V Loc N1esprit (binary)	Sentence with a subject denoting an idea and a locative complement denoting a person's mind ; the sentence has a metaphorical meaning Example : *L'idée d'essayer frétillait dans son esprit*
N0pc lui V W (binary)	Pronominalization of a noun which denotes a person and is a post-modifier of a subject denoting a body part ; the pronoun is a dative clitic Example : *Les yeux lui clignent ; La tête lui a percuté contre le dossier*
N1 = (N) (Adj) (binary)	A type of subject raising : a sentential complement N1 in the indicative mood taking the form que N être Adj W can be replaced with a constituent N followed by an objective complement Adj W Example : *J'imagine* **que Marie est enceinte** = *J'imagine* **Marie enceinte**
N1 = (N) (être Adj) (binary)	A type of subject raising : a sentential complement N1 in the indicative mood taking the form que N être Adj W can be replaced with a constituent N followed by an infinitive clause être Adj W Example : *J'imagine* **que Marie est ironique** = *J'imagine* **Marie être ironique**
N1 = (N) (V-ant W) (binary)	A type of subject raising : a sentential complement N1 in the indicative mood taking the form que N V W can be replaced with a constituent N followed by a participial clause V-ant W Example : *J'ai repéré* **que Paul travaille** = *J'ai repéré* **Paul travaillant**
N1 = (N) (V-inf W) (binary)	A type of subject raising : a sentential complement N1 in the indicative mood taking the form que N V W can be replaced with a constituent N followed by an infinitive clause V-inf W Example : *Je vois* **que Paul travaille** = *Je vois* **Paul travailler**
N1 = où (binary)	An object N1 answers a question in *où ?* Example : *Où le livre est-il paru ? - Il est paru chez Plon ; D'où l'eau filtre-t-elle ? - Elle filtre du plafond*
N1 = Ppv (binary)	A nominal object N1 can pronominalize into a clitic Example : *Paul aime Marie = Paul l'aime*
N1 =: Qu P = Aux V0-inf W (binary)	A direct sentential complement in the indicative mood noted N1, when its subject is coreferent to the main subject and it contains an auxiliary or some verbs (Gross, 1975) (p. 76), can be replaced by a non-prepositional infinitive clause with an implicit subject coreferent to the main subject Example : *Paul dit qu'il a dormi = Paul dit avoir dormi ; Paul dit qu'il ignore l'anglais = Paul dit ignorer l'anglais (cf. Paul dit qu'il se promène, *Paul dit se promener) ; Paul inscrit dans l'agenda qu'il a fait ça = Paul inscrit dans l'agenda avoir fait ça*

Table E.23: Documentation des propriétés des tables de verbes (23)

Feature	Description of the feature
N1 =: ce(ci+la) (binary)	A sentential complement N1 can pronominalize in the form of *ceci* or *cela* Example : *Cet échec équivaut pour Max à ce qu'il doive changer d'objectif = Cet échec équivaut pour Max à (ceci+cela)*
N1 =: combien P (binary)	An object N1 can take the form of an exclamative clause Example : *Max contemple combien tout est beau*
N1 =: coup (binary)	An object N1 can denote a blow Example : *Max allonge une claque à Léa ; Paul reçoit un coup de pied de Jean*
N1 =: de V0-inf W (binary)	A direct object N1 can take the form of an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to the main subject Example : *Paul subordonne de faire ça au résultat des élections*
N1 =: de V2c-inf W (binary)	A direct object N1 can take the form of an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to a nominal post-modifier of a nominal object N2 Example : *Paul incorpore dans les instructions de Luc de faire la vaisselle*
N1 =: de V2-inf W (binary)	A direct object N1 can take the form of an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to an object N2 Example : *Ceci justifie pour Paul de démissionner*
N1 =: Dnum Nmes (binary)	An object N1 can take the form of a cardinal numeral and a noun of measurement unit Example : *Ce vin titre 13 degrés*
N1 =: Dnum Nmonnaie (binary)	An object N1 can take the form of a cardinal numeral and a noun of measurement unit, denote a sum of money and answer a question in *combien ?* Example : *Paul a avancé 100 francs à Jean ; Combien Paul a-t-il avancé à Jean ?*
N1 =: esprit, N2 =: idée (binary)	An object N1 can denote a person's mind and an object N2 an idea ; the sentence has a metaphorical meaning Example : *Paul embarrasse son esprit d'idées saugrenues ; Luc débarrasse sa tête de ces projets idiots*
N1 =: idée, (N2+N3) =: esprit (binary)	An object N1 can denote an idea, and an object noted N2 or N3, a person's mind ; the sentence has a metaphorical meaning with regard to the locative meaning Example : *Léa a évacué ces idées de l'esprit de Luc*
N1 =: idée, N2 =: esprit (binary)	An object N1 can denote an idea and an object N2 a person's mind ; the sentence has a metaphorical meaning Example : *Paul a fourré de drôles d'idées dans la tête de Luc*
N1 =: le fait Qu P (binary)	An object N1 can take the form of *le fait que P*, close to a sentential complement, and denote an abstract entity Example : *Max a décortiqué le fait que Paul ne soit pas venu*
N1 =: mot, (N2+N3) =: texte (binary)	An object N1 can denote words, and an object noted N2 or N3, a text ; the sentence has a metaphorical meaning with regard to the locative meaning Example : *Paul bannit les grossièretés de ses articles ; Paul a centralisé les informations dans son article*

TABLE E.24: Documentation des propriétés des tables de verbes (24)

Feature	Description of the feature
N1 =: mot, N2 =: texte (binary)	An object N1 can denote words and an object N2 a text ; the sentence has a metaphorical meaning Example : *Paul a introduit des citations bizarres dans son article ; Paul saupoudre des plaisanteries dans ses discours*
N1 =: Nabs (binary)	An object N1 can denote an abstract entity Example : *Paul dispense des conseils ; Paul attribue de l'importance à ce que dit Léa*
N1 =: Nabs métaphore (binary)	An object N1 can denote an abstract entity ; the sentence has a metaphorical meaning (N1 normally denotes obligatorily a person or concrete object) Example : *Ces contraintes asphyxient mon enthousiasme ; Paul a abîmé son amour-propre*
N1 =: Nconc (binary)	An object N1 can denote a concrete object Example : *Le radar suit la fusée*
N1 =: Nconc métaphore (binary)	An object N1 can denote a concrete object ; the sentence has a metaphorical meaning (N1 normally denotes obligatorily a person) Example : *Paul rapatrie ses capitaux en France*
N1 =: Nhum (binary)	An object N1 can denote a person Example : *Cette robe va très bien à Marie ; On a vacciné Paul contre le tétanos ; Léa confie son fils à des voisins ; Ida rassasie son chat de poisson*
N1 =: N-hum (binary)	An object N1 can denote an entity which is not a person Example : *Luc court après* **l'argent** *; Max ampute* **sa thèse** *de deux chapitres*
N1 =: Npc (binary)	An object N1 can denote a body part Example : *Paul joint le pouce avec l'index*
N1 =: Npc obl (binary)	An object N1 denotes obligatorily a body part, or, metonymically, a person Example : *Le soleil a basané la peau de Max = Le soleil a basané Max*
N1 =: Npl obl (binary)	An object N1 is obligatorily in the plural or has a collective interpretation Example : *La bouteille a éclaté en mille morceaux*
N1 =: Qu P (binary)	An object N1 can take the form of a sentential complement, in a sentence with a metaphorical meaning Example : *On exhume de manuscrits grecs que la Terre est ronde*
N1 =: Qu P = Ppv (binary)	An object N1 taking the form of a sentential complement can pronominalize into a clitic Example : *Paul a combiné avec Luc que Marie soit écartée = Paul l'a combiné avec Luc ; Luc va empêcher que Marie vienne = Luc va l'empêcher*
N1 =: Qu Pind (binary)	An object N1 can take the form of a sentential complement in the indicative mood. When N1 is introduced by a preposition, the particle *ce* appears between the preposition and the complementizer *que* Example : *Paul instaure que les réunions auront lieu à 10 h ; Ceci découle de ce que Luc est revenu*
N1 =: Qu Psubj (binary)	An object N1 can take the form of a sentential complement in the subjunctive mood. When N1 is introduced by a preposition, the particle *ce* appears between the preposition and the complementizer *que* Example : *Paul instaure que les réunions aient lieu à 10 h ; Luc a contribué à ce que Paul soit élu*

TABLE E.25: Documentation des propriétés des tables de verbes (25)

Feature	Description of the feature
N1 =: si P ou si P (binary)	An object N1 can take the form of an interrogative clause Example : *Paul annoncera à tous si Léa vient ou pas*
N1 =: texte, N2 =: mot (binary)	An object N1 can denote a text and an object N2 words ; the sentence has a metaphorical meaning Example : *Paul débarasse l'article de tous les jeux de mots ; Max truffe sa thèse de citations*
N1 =: trou (binary)	An object N1 can denote a hole Example : *Paul a ménagé une ouverture dans la paroi ; Paul a découpé un carré dans la feuille*
N1 =: V0-inf W (binary)	An object N1 can take the form of an infinitive clause, with an implicit subject coreferent to the main subject Example : *Paul pense avoir bien fait ; Luc cherche à partir loin*
N1 =: V2-inf W (binary)	An object N1 can take the form of an infinitive clause, with an implicit subject coreferent to an object N2 Example : *Le travail consiste pour Paul à recevoir les clients*
N1 apparition (binary)	An object N1 is interpreted as appearing or being created during the process Example : *Max a bâti une maison*
N1 bénéficiaire (binary)	An object N1 is interpreted as receiving the referent of N2 as a benefit Example : *Max goinfre Bob de gâteaux*
N1 détrimentaire (binary)	An object N1 is interpreted as losing the referent of N2 Example : *Max a possédé Luc de 100 euros*
N1 disparition (binary)	An object N1 is interpreted as disappearing during the process Example : *Max a démoli la maison*
N1 mouvement (binary)	An object N1 is interpreted as a moving object or place Example : *La voiture dépasse le camion*
N1 en V (binary)	In a sentence of the form N1 V de N0, the object N0 can pronominalize into the clitic pronoun *en* Example : *Cette région regorge de pétrole = Cette région en regorge*
N1 et N2 faire V-n (binary)	A conjunction of the objects N1 and N2 can be subject of faire followed by a deverbal noun morphologically associated with V Example : *Paul mélange les deux liquides = Les deux liquides font un mélange*
N1 et N2 faire V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a conjunction of the objects N1 and N2 can be subject of faire followed by the deverbal noun Example : *mélange*
N1 être V-adj à N2 (binary)	An object N1 can become the subject of a sentence with *être* followed by a deverbal adjective morphologically associated with V, and by an object à N2 with the same distribution as in the verbal sentence Example : *On a imperméabilisé ce tissu à l'eau = Ce tissu est imperméable à l'eau*
N1 être V-adj de N2 (binary)	An object N1 can become the subject of a sentence with *être* followed by a deverbal adjective morphologically associated with V, and by an object de N2 with the same distribution as in the verbal sentence Example : *On a libéré Paul de ses liens = Paul est libre de ses liens ; ?On a noirci la pièce de toute cette suie = ?La pièce est noire de toute cette suie (as opposed to : On a allégé le paquet de 100 g/*Le paquet est léger de 100 g)*

TABLE E.26: Documentation des propriétés des tables de verbes (26)

Feature	Description of the feature
N1 être V-ant (binary)	An object N1 can become the subject of a sentence with *être* followed by a deverbal adjective morphologically associated with V, with the suffix *-ant* ; other objects are not preserved Example : *Le papier est jaunissant ; Le gibier est abondant*
N1 être V-ant Prép N0 (binary)	An object N1 can become the subject of a sentence with *être* followed by a deverbal adjective morphologically associated with V, with the suffix *-ant*, and by an object Prép N0 Example : *Les étoiles sont étincelantes dans le ciel = Le ciel étincelle d'étoiles = Le ciel est étincelant d'étoiles*
N1 être V-n (binary)	An object N1 can become the subject of a sentence with *être* and a deverbal noun morphologically associated with V as a 'predicate nominative' Example : *Paul a torchonné son article = Son article est un torchon*
N1 être V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used when an object N1 becomes the subject of a sentence with *être* followed by the deverbal noun as a 'predicate nominative' Example : *torchon*
N1 être V-n pour N0 (binary)	An object N1 can become the subject of a sentence with *être* followed by a deverbal noun morphologically associated with V, and by an object pour N0 Example : *Paul abomine cette situation = Cette situation est une abomination pour Paul*
N1 être V-n pour N0 (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used when an object N1 becomes the subject of a sentence with *être* followed by the deverbal noun and by an object pour N0 Example : *abomination*
N1 être V-n, Sfx # E (binary)	An object N1 can become the subject of a sentence with *être* followed by a deverbal noun morphologically associated with V, with a non-empty suffix Example : *Le volcan projette de la cendre = La cendre est une projection ; Luc bave sur la serviette = Luc fait une bavure sur la serviette*
N1 être V-n, Sfx # E (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used when an object N1 becomes the subject of a sentence with *être* followed by the deverbal noun, with a non-empty suffix Example : *bavure*
N1 être Vpp (binary)	Passive without complements, acceptable in the present tense and interpreted as a static result Example : *Luc balaie le couloir de tous les détritus = Le couloir est balayé ; Luc ancre le bateau dans la crique = Le bateau est ancré ; Le choc éboule le mur en gravats = Le mur est éboulé*
N1 être Vpp de ce Qu P (binary)	Passivization of a sentence with a sentential subject ; the agent is introduced by the preposition *de* ; other objects, if any, are not preserved ; the sentence is acceptable in the present tense and interpreted as a static result Example : *Que Luc s'en aille étonne Paul = Paul est étonné de ce que Luc s'en aille ; Que Marie soit venue excite Pierre à se faire valoir = Pierre est excité de ce que Marie soit venue*

TABLE E.27: Documentation des propriétés des tables de verbes (27)

Feature	Description of the feature
N1 être Vpp de N0 (binary)	Passivization with agent introduced by the preposition *de* ; other objects, if any, are not preserved ; if N1 is a prepositional object, it becomes the passive subject without its preposition Example : *Tous détestent Paul = Paul est détesté de tous ; Tous ont discuté de ce problème = Ce problème a été discuté de tous*
N1 être Vpp de N0 à ce Qu P (binary)	Passivization with agent introduced by the preposition *de* ; an object à ce Qu P is preserved Example : *Paul a été encouragé de tous à faire ça*
N1 être Vpp de N0 de ce Qu P (binary)	Passivization with agent introduced by the preposition *de* ; an object de ce Qu P is preserved Example : *Paul a été hué de la foule d'avoir fait ça*
N1 être Vpp de N0 Prép N2 (binary)	Passivization with agent introduced by the preposition *de* ; an object Prép N2 is preserved ; if N1 is a prepositional object, it becomes the passive subject without its preposition Example : *Tous ont lu la nouvelle dans le journal = La nouvelle a été lue de tous dans le journal ; Tous ont attesté de ce fait auprès des enquêteurs = Ce fait a été attesté de tous auprès des enquêteurs*
N1 être Vpp par N0 (binary)	Passivization with agent introduced by the preposition *par* ; other objects are not preserved ; if N1 is a prepositional object, it becomes the passive subject without its preposition Example : *Tous détestent Paul = Paul est détesté par tous ; Tous ont discuté de ce problème = Ce problème a été discuté par tous*
N1 être Vpp par N0 à ce Qu P (binary)	Passivization with agent introduced by the preposition *par* ; an object à ce Qu P is preserved Example : *Paul a été encouragé par tous à faire ça*
N1 être Vpp par N0 de ce Qu P (binary)	Passivization with agent introduced by the preposition *par* ; an object de ce Qu P is preserved Example : *Paul a été hué par la foule d'avoir fait ça*
N1 être Vpp par N0 Prép N2 (binary)	Passivization with agent introduced by the preposition *par* ; an object Prép N2 is preserved ; if N1 is a prepositional object, it becomes the passive subject without its preposition Example : *Tous ont lu la nouvelle dans le journal = La nouvelle a été lue par tous dans le journal ; Tous ont attesté de ce fait auprès des enquêteurs = Ce fait a été attesté par tous auprès des enquêteurs*
N1 être Vpp W (binary)	Passivization without agent ; prepositional objects, if any, are preserved ; the sentence is acceptable in the present tense and interpreted as a static result Example : *Le mot chien est traduit par le mot dog ; Ce pain est grillé (cf. *Cet oeuf est gobé)*
N1 lieu de destination de N0 (binary)	An object N1 is interpreted as a destination of the referent of N0. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Max atteint le sommet*
N1 lieu de destination de N2 (binary)	An object N1 is interpreted as a destination of the referent of N2. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Luc sale le rôti de sel fin*

TABLE E.28: Documentation des propriétés des tables de verbes (28)

Feature	Description of the feature
N1 lieu de passage de N0 (binary)	An object N1 is interpreted as a place where the referent of the subject passes ; the sentence is interpreted with a punctual aspect Example : *Le car emprunte l'avenue*
N1 lieu source de N0 (binary)	An object N1 is interpreted as the source of the referent of N0. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *L'ennemi abandonne la ville*
N1 lieu source de N2 (binary)	An object N1 is interpreted as the source of the referent of N2. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Max cure le puits de sa vase*
N1 lieu statique de N0 (binary)	An object N1 is interpreted as a place where the referent of the subject is before, during and after the process. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Léa arpente le boulevard*
N1 se V (neutre) (binary)	An object N1 can become the subject of a sentence with a pronominal verb and no other objects ; the reflexive clitic is in the accusative ; the sentence is not the result of the clitic pronominalization of an object N1 coreferent to N0 ; the sentence does not imply the existence of a human agent ; the sentence denotes a datable event (neutral construction in the sense of Ruwet (1972)) Example : *Le choc a ankylosé son bras = Son bras s'ankylose (as opposed to : On entretient cette voiture facilement = Cette voiture s'entretient facilement)*
N1 se V auprès de N3hum de ce Qu P (binary)	An object N1 denoting a person can become the subject of a sentence with a pronominal verb, an object of the form de ce que P, which is either N0 or N2, and an additional object of the form auprès de N3, denoting a person ; the sentence implies a verbal communication between the two persons ; the reflexive clitic is in the accusative Example : *Que Léa ne soit pas venue attriste Paul = Paul s'attriste auprès de Marie de ce que Léa ne soit pas venue ; Quelque chose réconforte Paul de ce qu'il a dû faire cela = Paul se réconforte auprès de Marie de ce qu'il a dû faire cela*
N1 se V de ce Qu P (binary)	An object N1 can become the subject of a sentence with a pronominal verb and object of the form de ce que P, which is either N0 or N2 ; the sentence may imply that N1 denotes a person which communicates verbally with another person, or not ; the reflexive clitic is in the accusative ; the sentence is not the result of the clitic pronominalization of an object N1 coreferent to N0 Example : *Que Max dorme choque Marie = Marie se choque de ce que Max dort ; Que Marie dorme exacerbe l'ire de Max = L'ire de Max s'exacerbe de ce que Marie dort ; Quelque chose console Luc de ce qu'il n'a pas été élu = Luc se console de ce qu'il n'a pas été élu*

TABLE E.29: Documentation des propriétés des tables de verbes (29)

Feature	Description of the feature
N1 se V en N2 (statique) (binary)	An object N1 can become the subject of a sentence with a pronominal verb and an object of the form en N2 ; the reflexive clitic is in the accusative ; the sentence is not the result of the clitic pronominalization of an object N1 coreferent to N0 ; the sentence does not imply the existence of a human agent ; the sentence denotes a stative situation Example : *On divise le terrain en trois zones = Le terrain se divise en trois zones (as opposed to : Le choc casse le caillou en trois morceaux = Le caillou se casse en trois morceaux)*
N1 V (binary)	An object N1 can become the subject of a sentence with no other objects Example : *Paul chauffe la soupe = La soupe chauffe ; Luc gonfle le ballon d'air = Le ballon gonfle (E+*d'air)*
N1 V avec N2 (binary)	An object N1 can become the subject of a sentence with an object avec N2 Example : *Le juge a divorcé Ida d'avec Luc = Ida a divorcé avec Luc*
N1 V de ce Qu Pind (binary)	An object N1 can become the subject of a sentence while the subject, which takes the form of a sentential subject, becomes an object of the form de ce que Pind, in the indicative mood Example : *Que Paul est vainqueur retentit dans toute la ville = Toute la ville retentit de ce que Paul est vainqueur*
N1 V de N0 source (binary)	An object N1 can become the subject of a sentence while the locative subject, which is interpreted as a source, becomes an object introduced by the preposition *de* Example : *La lampe irradie une lumière douce = Une lumière douce irradie de la lampe*
N1 V de N1pc (binary)	When a locative object, denoting a body part of a person, takes the form Loc N1pc de N1, where N1 denotes the person, a sentence can have N1 as its subject and de N1pc as an object Example : *Des bruits incongrus glougloutent dans l'intestin de Jean = Jean glougloute de l'intestin*
N1 V Loc N2 (binary)	An object N1 can become the subject of a sentence with an object Loc N2 ; the sentence is interpreted either as processive or as a static result Example : *Max approche la bouteille du verre = La bouteille approche du verre ; Max tient le carton contre la caisse = Le carton tient contre la caisse*
N1 V N2 (binary)	An object N1 can become the subject of a sentence with a direct object N2 ; the sentence is interpreted as processive Example : *Je cogne mon verre contre le tien = Mon verre cogne le tien*
N1 V Prép N0 (binary)	A locative object N1, with a scenic interpretation, can become the subject of a sentence, while the subject becomes an object introduced by a preposition ; some changes in determiners may take place Example : *Les abeilles grouillent dans le jardin = Le jardin grouille d'abeilles ; Les pins abondent dans la forêt = La forêt abonde en pins*
N1 V Prép N0 = N0 être V-n (binary)	Sentence with noun-centered argument structure, with the same subject, the verb *être*, and a deverbal noun related with V ; the distribution of N0 in the sentence is a subset of its distribution in N1 V Prép N0 Example : *Le salon froufroute du bruit des soieries = Le bruit des soieries est un froufrou (cf. Les soieries froufroutent dans le salon = Le salon froufroute de soieries, *Les soieries sont un froufrou)*

Table E.30: Documentation des propriétés des tables de verbes (30)

Feature	Description of the feature
N1 V Prép N0hum (binary)	In N1 V Prép N0, the object N0 may denote persons Example : *La rue déborde de touristes (cf. *Les touristes débordent dans la rue)*
N1 V Prép N0pl obl (binary)	In N1 V Prép N0, the object N0 is obligatorily in the plural or has a collective interpretation Example : *Des rires moqueurs fusèrent dans la salle = La salle fusa de rires moqueurs (cf. Un rire moqueur fusa dans la salle, *La salle fusa d'un rire moqueur)*
N1 V W (binary)	An object N1 can become the subject of a sentence with conservation of its other objects, if any Example : *Max enfonce le bâton dans le sable = Le bâton enfonce dans le sable*
N1c =: Nhum (binary)	A nominal direct object of the form N1 de N1c, where N1c denotes a person, can be replaced by N1c and a prepositional modifier Prép N1 called 'echo modifier' (restructuration) Example : *Ceci confirme la décision de Luc = Ceci confirme Luc dans sa décision*
N1c =: N-hum (binary)	A nominal direct object of the form N1 de N1c, where N1c denotes a non-human entity, can be replaced by N1c and a prepositional modifier Prép N1 called 'echo modifier' (restructuration) Example : *Ceci freine l'élan de l'avalanche = Ceci freine l'avalanche dans son élan*
N1c être Vpp (binary)	A post-modifier of a direct object N1 can become the subject of a sentence with *être* and the past participle of V ; the sentence has a stative interpretation Example : *Ceci a ébranlé les certitudes de Luc = Luc est ébranlé*
N1-hum = Ppv (binary)	An object N1 denoting a non-human entity can be pronominalized into a clitic pronoun Example : *Luc suspend sa décision à une condition = Luc la suspend à une condition*
N1hum V Prép N0 (binary)	In N1 V Prép N0, the subject N1, with a scenic interpretation, may denote a person Example : *Marie rutile de diamants*
N1pc lui V (binary)	When an object, denoting a body part of a person, takes the form N1pc de N1, where N1 denotes the person, a sentence can have N1pc as its subject and a dative clitic coreferent to N1 Example : *Le froid hérisse les poils de Luc = Les poils lui hérissent ; De la sueur suinte sur le front de Luc = Le front lui suinte*
N2 =: Adj (binary)	An object N2 can take the form of an adjective alongwith its objects and modifiers, if any Example : *Paul proclame Luc digne de ce titre*
N2 =: ce(ci+la) (binary)	A sentential complement N2 can pronominalize in the form of *ceci* or *cela* Example : *Paul habitue Luc à ce qu'il n'y ait pas de déjeuner = Paul habitue Luc à cela*
N2 =: couche (binary)	An object N2 can be interpreted as a layer of a concrete substance Example : *Luc badigeonne le mur (E+d'une couche) de chaux ; Luc encolle le papier (E+d'une pellicule) de colle*

Table E.31: Documentation des propriétés des tables de verbes (31)

Feature	Description of the feature
N2 =: coup (binary)	An object N2 can be interpreted as a blow Example : *Luc martèle la table de coups de poing ; Luc scande son chant de claquements de mains*
N2 =: déformation (binary)	An object N2 can denote the type of distorsion underwent by the object N1 Example : *Paul a cabossé le plateau de grosses bosses*
N2 =: Dnum Nmes (binary)	A direct complement N2 can take the form of a cardinal numeral followed by a noun of measurement unit Example : *Paul cote ce livre 100 euros*
N2 =: le fait Qu P (binary)	An object N2 can take the form le fait que P, close to a sentential complement, and denote an abstract entity Example : *Paul avertit Luc du fait que Léa arrive*
N2 =: N de V-n (binary)	The object N2 is of the form N de V-n, where V-n is a deverbal noun related to V Example : *On a empierré la route de (E+une couche de) pierres*
N2 =: Nhum (binary)	An object N2 can denote a person Example : *Luc éloigne Paul de Marie ; On évacue la salle de tous les gêneurs ; Le général assiège la ville de troupes fraîches*
N2 =: N-hum (binary)	An object N2 can denote an entity which is not a person Example : *On a débarrassé la cave de toutes les bouteilles ; On a attribué le prix à cette oeuvre*
N2 =: Npc (binary)	An object N2 can denote a body part Example : *Paul joint le pouce avec l'index*
N2 =: Npl obl (binary)	An object N2 is obligatorily in the plural or has a collective interpretation Example : *Luc distribue des bonbons (*à l'+aux) enfants ; Luc émaille son récit (?*d'une+de) citations*
N2 =: Npr (binary)	An object N2 can take the form of a proper name Example : *Paul a baptisé sa fille Marie*
N2 =: point (binary)	An object N2 can have as its head noun the noun point interpreted as an abstract entity Example : *Paul subdivise son chapitre en plusieurs points*
N2 =: Qu Pind (binary)	A prepositional object N2 can take the form of a sentential complement in the indicative mood. The particle *ce* appears between the preposition and the complementizer *que* Example : *Max hue le gardien de ce qu'il a raté la balle*
N2 =: Qu Psubj (binary)	A prepositional object N2 can take the form of a sentential complement in the subjunctive mood. The particle *ce* appears between the preposition and the complementizer *que*, except if the object is introduced by *pour*, *avec* or *et* Example : *Ceci incite Zoé à ce que tout soit en ordre ; Max compte sur Luc pour que tout soit en ordre ; Luc alterne que le fil rouge passe sur le bleu et qu'il passe dessous*
N2 =: si P ou si P (binary)	An object N2 can take the form of an interrogative clause Example : *Paul avisera Luc si Léa vient ou pas*
N2 =: trace (binary)	An object N2 can denote marks Example : *Paul a paginé ses feuilles de numéros fantaisistes*
N2 =: trou (binary)	An object N2 can denote a hole Example : *Luc composte le ticket d'un petit trou carré ; Luc entaille le bâton d'une entaille profonde*

TABLE E.32: Documentation des propriétés des tables de verbes (32)

Feature	Description of the feature
N2 =: V-n de N (binary)	The object N2 is of the form V-n de N, where V-n is a deverbal noun related to V Example : *Paul assaisonne son plat de (E+un assaisonnement de) épices exotiques*
N2 =: V0-inf W (binary)	An object N2 can take the form of an infinitive clause, with an implicit subject coreferent to the main subject Example : *Paul menace Léa de la quitter*
N2 =: V1c-inf W (binary)	An object N2 can be an infinitive clause with an implicit subject coreferent with a nominal post-modifier of a nominal object N1 Example : *La situation limite l'activité de Paul à amuser Marie*
N2 =: V1-inf W (binary)	An object N2 can take the form of an infinitive clause, with an implicit subject coreferent to an object N1 Example : *Ceci a réduit Luc à mendier*
N2 =: zone (binary)	An object N2 can denote a zone Example : *Paul a morcelé son terrain en plusieurs zones*
N2 apparition (binary)	An object N2 is interpreted as appearing or being created during the process Example : *Le soleil a basané sa peau d'un hâle cuivré ; On a bruité ce court métrage de bruits désagréables*
N2 bénéficiaire (binary)	An object N2 is interpreted as receiving the referent of N1 as a benefit Example : *On a accordé un prêt à Luc*
N2 destination V N1 (binary)	An object N2, interpreted as a destination, can become the subject of a sentence with a direct object N1 ; the sentence is interpreted as a static result Example : *Paul rassemble les informations dans un fichier = Un fichier rassemble les informations*
N2 détrimentaire (binary)	An object N2 is interpreted as losing the referent of N1 Example : *Luc hérite une horloge de sa grand-mère ; Luc a acheté cet immeuble à Marie*
N2 être V-n (binary)	Sentence with noun-centered argument structure, with an object N2 as the subject, the verb *être*, and a deverbal noun related with V Example : *On a monétisé ce métal en pièces = Les pièces sont de la monnaie ; Max a affermé sa terre à Luc = Luc est fermier ; On a démoulé le gâteau de cet ustensile = Cet ustensile est un moule*
N2 être V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a sentence with noun-centered argument structure, with an object N2 as the subject, the verb *être*, and the deverbal noun related Example : *monnaie*
N2 être V-n, Sfx # E (binary)	Sentence with noun-centered argument structure, with an object N2 as the subject, the verb *être*, and a deverbal noun related with V with a non-empty suffix Example : *Luc accoutre Léa d'une tenue invraisemblable = Cette tenue est un accoutrement ; On inonde la région d'une eau boueuse = Cette eau boueuse est une inondation ; On pèle les fruits de leur peau = Leur peau est une pelure*

TABLE E.33: Documentation des propriétés des tables de verbes (33)

Feature	Description of the feature
N2 être V-n, Sfx = E (binary)	Sentence with noun-centered argument structure, with an object N2 as the subject, the verb *être*, and a deverbal noun related with V with an empty suffix Example : *Luc débroussaille l'allée de ces ronces = Ces ronces sont des broussailles ; On a asphalté la route de ce revêtement = Ce revêtement est de l'asphalte*
N2 lieu de destination de N0 (binary)	An object N2 is interpreted as the destination of the referent of the subject. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Le bus débouche de la rue sur la place*
N2 lieu de destination de N1 (binary)	An object N2 is interpreted as the destination of the referent of N1. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Paul souffle de l'air dans le ballon*
N2 lieu source de N1 (binary)	An object N2 is interpreted as the source of the referent of N1. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *La grue a levé le bloc du sol*
N2 lieu statique de N1 (binary)	An object N2 is interpreted as a place where the referent of an object N1 is before, during and after the process. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Bob cuit la porcelaine au four*
N2 scénique être V-n (binary)	A locative modifier N2, with a scenic interpretation, can become the subject of a sentence with *être* and a deverbal noun related with V as a 'predicate nominative' Example : *Paul forge des objets dans cet atelier = Cet atelier est une forge*
N2 scénique être V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used when a locative modifier N2, with a scenic interpretation, becomes the subject of a sentence with *être* and the deverbal noun as a 'predicate nominative' Example : *forge*
N2 V N1 (binary)	An object N2 can become the subject of a sentence with a direct object N1 ; the sentence is interpreted as a static result Example : *Luc assaisonne la salade de mayonnaise = De la mayonnaise assaisonne la salade ; Luc dissimule le sac derrière le rideau = Le rideau dissimule le sac ; On a représenté la chapelle par une croix = Une croix représente la chapelle*
N3 =: ce(ci+la) (binary)	A sentential complement N3 can pronominalize in the form of *ceci* or *cela* Example : *Paul paye une grosse somme à Luc pour que la lettre arrive à temps = Paul paye une grosse somme à Luc pour cela*
N3 =: de V1-inf W (binary)	An object N3 can take the form of an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to an object N1 Example : *Paul donne à Luc comme travail de faire ça*
N3 =: le fait Qu P (binary)	An object N3 can take the form le fait que P, close to a sentential complement, and denote an abstract entity Example : *On défend le métal de la corrosion par le fait que les embruns ne peuvent plus l'atteindre*

Table E.34: Documentation des propriétés des tables de verbes (34)

Feature	Description of the feature
N3 =: Nhum (binary)	An object N3 can denote a person Example : *Max défend la ville de bastions contre l'ennemi*
N3 =: N-hum (binary)	An object N3 can denote an entity which is not a person Example : *Paul désigne Luc à Léa pour cette corvée*
N3 =: Npl obl (binary)	An object N3 is obligatorily in the plural or has a collective interpretation Example : *Paul répartit les élèves dans quatre classes*
N3 =: Qu Pind (binary)	An object N3 can take the form of a sentential complement in the indicative mood Example : *Paul mise cet argent sur Luc qu'il va gagner la course*
N3 =: Qu Psubj (binary)	An object N3 can take the form of a sentential complement in the subjunctive mood Example : *Paul sert à Luc de complice pour qu'il fasse ça*
N3 =: V0-inf W (binary)	An object N3 can take the form of an infinitive clause, with an implicit subject coreferent to the main subject Example : *J'obtiens une aide de Luc pour me soigner*
N3 =: V1-inf W (binary)	An object N3 can take the form of an infinitive clause, with an implicit subject coreferent to an object N1 Example : *Ces querelles servent de prétexte à Luc pour sortir*
N3 =: V2-inf W (binary)	An object N3 can take the form of an infinitive clause, with an implicit subject coreferent to an object N2 Example : *Paul donne du temps à Luc pour faire ça*
N3 être V-n (binary)	Sentence with noun-centered argument structure, with an object N3 as the subject, the verb *être*, and a deverbal noun related with V Example : *Paul a classé les verbes dans ces dix groupes = Ces dix groupes sont des classes*
N3 être V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a nominal sentence with an object N2 as the subject, the verb *être*, and the deverbal noun related Example : *classe*
N3 attache V N1 (binary)	A complement N3 attache denoting a tying system and interpreted as an instrument can become the subject of a sentence with a direct object N1 ; the sentence is interpreted as a static result Example : *Luc ligature les lèvres de la plaie d'un fil rouge = Un fil rouge ligature les lèvres de la plaie*
N3 lieu de destination de N1 (binary)	An object N3 is interpreted as the destination of the referent of N1. The criterion to be applied is that of support sentences as in Boons *et al.* (1976b) (p. 205) and in Guillet et Leclère (1992) (p. 22) Example : *Luc tracte la voiture de Gap à Dax*
N actif V N0 (binary)	The subject of the basic construction can be a direct object in a sentence with a subject denoting a person which is interpreted as active Example : *(La table mesure 3 mètres) Paul mesure la table*
Neg (binary)	A negative adverb is obligatorily present Example : *Luc n'arrête pas d'être dérangé*
où N0 V-il ? (binary)	In a sentence with a non-prepositional infinitive clause which has an implicit subject coreferent with that of the main verb, the infinitive clause answers a question with *où* and no other object Example : *Où Paul court-il ? - Il court chercher du vin*

TABLE E.35: Documentation des propriétés des tables de verbes (35)

Feature	Description of the feature
où N0 V-il N1 ? (binary)	In a sentence with a non-prepositional infinitive clause which has an implicit subject coreferent with that of the main verb, the infinitive clause answers a question with *où* and a direct object N1 Example : *Où Luc emmène-t-il Marie ? - Il emmène Marie voir un film*
'P', V N0 à N2 (binary)	The sentential complement N1 can take the form of direct speech and appear in the beginning of the sentence ; the subject is after the verb Example : *"Je vais au cinéma", répond Luc à Léa*
par N4 (binary)	Additional locative object introduced by the preposition *par* and interpreted as a place of passage Example : *Luc passe le lit de la chambre dans le salon par le couloir*
Ppv =: en (binary)	Clitic pronominalization of an object introduced by the preposition *de* into *en* Example : *Max a dépouillé Ida de ses biens = Max en a dépouillé Ida ; L'au gicla du tonneau = L'eau en gicla ; Luc reçoit un colis de Marie = Luc en reçoit un colis*
Ppv =: en figé (binary)	The clitic pronoun *en* is frozen with the verb V Example : *Je n'en reviens pas de ce culot ; Luc s'en va*
Ppv =: la figé (binary)	The clitic pronoun *la* is frozen with the verb V Example : *Fermez-la*
Ppv =: le (binary)	Clitic pronominalization of a direct object N1 into *le* Example : *Paul compare Luc à un singe = Paul le compare à un singe ; Luc a bâti cette maison = Luc l'a bâtie ; On a élu Luc président = On l' a élu président*
Ppv =: le figé (binary)	The clitic pronoun *en* is frozen with the verb V Example : *Je n'en reviens pas de ce culot ; Luc s'en va*
Ppv =: les figé (binary)	The clitic pronoun *les* is frozen with the verb V Example : *Max les aligne à Luc*
Ppv =: lui (binary)	Clitic pronominalization of an object introduced by the preposition *à* into *lui* Example : *Luc obéit à Léa = Luc lui obéit ; Paul consacre de l'argent à son jardin = Paul lui consacre de l'argent*
Ppv =: Neg (binary)	The clitic pronoun *ne* is frozen with the verb V, even without any adverb of negation Example : *Luc n'arrête pas d'être dérangé ; Luc ne saurait dormir ici*
Ppv =: se figé (binary)	The clitic pronoun *se* is frozen with the verb V Example : *Luc se réserve pour la nuit ; Luc s'en va*
Ppv =: y (binary)	Clitic pronominalization of an object introduced by a preposition into *y* Example : *Luc obéit à cette loi = Luc y obéit ; Luc punaise les notes sur le tableau = Luc y punaise les notes*
Ppv =: y figé (binary)	The clitic pronoun *y* is frozen with the verb V Example : *Luc y va ; Luc s'y croit*
Prép N1 = Ppv (binary)	A nominal object Prép N1 can pronominalize into a clitic ; the preposition can be empty Example : *Paul est près d'une catastrophe = Paul en est près*

TABLE E.36: Documentation des propriétés des tables de verbes (36)

Feature	Description of the feature
Prép N1 =: à Nq (binary)	An object Prép N1 can have à as its preposition and an abstract noun as its head noun Example : *Il parvient à la conscience de Max que Luc était son ami*
Prép N1 =: dans Nq (binary)	An object Prép N1 can have dans as its preposition and an abstract noun as its head noun Example : *Il intervient dans les négociations que Marie va partir*
Prép N1 =: de Nq (binary)	An object Prép N1 can have de as its preposition and an abstract noun as its head noun Example : *Il est sorti de cette discussion que Luc fera du yoga*
Prép N1 =: Prép ce Qu P = Qu P (binary)	Preposition and ce before a sentential complement N1 in que can be erased Example : *Luc s'aperçoit de ce que Marie l'a trompé à ce qu'elle est enceinte = Luc s'aperçoit que Marie l'a trompé à ce qu'elle est enceinte*
Prép N1 =: Prép Qu P = Ppv (binary)	A sentential complement Prép N1 can pronominalize into a clitic ; the preposition can be empty Example : *Paul parvient à ce que Luc l'aide = Paul y parvient ; Paul doute de ce que Luc puisse réussir = Luc en doute ; Luc doit que Marie reste à son physique = Luc le doit à son physique*
Prép N1hum = Ppv (binary)	An object Prép N1, denoting a person, can pronominalize into a clitic ; the preposition can be empty Example : *Faire ce travail plaît à Paul = Faire ce travail lui plaît ; Luc se méfie de Paul = Luc s'en méfie ; Luc préfère Marie à Léa = Luc la préfère à Léa*
Prép N1-hum = Ppv (binary)	An object Prép N1, denoting a non-human entity, can pronominalize into a clitic ; the preposition can be empty Example : *Luc a survécu à cette épreuve = Luc y a survécu ; Paul parle de ça avec Luc = Paul en parle avec Luc ; Luc préfère le vin à l'eau = Luc le préfère à l'eau*
Prép N2 = Prép là (binary)	A sentential complement Prép N2 can be pronominalized into Prép là ; if the preposition is *à*, it is erased Example : *Max a extrapolé de ce que Léa était absente que Luc était revenu = Max a extrapolé de là que Luc était revenu ; Max ramène l'inflation à ce que la production a diminué = Max ramène là l'inflation*
Prép N2 =: à Nq (binary)	An object Prép N2 can have à as its preposition and an abstract noun as its head noun Example : *Il intègre à son programme de partir à 3 h*
Prép N2 =: dans Nq (binary)	An object Prép N2 can have dans as its preposition and an abstract noun as its head noun Example : *Il inclut dans son programme de partir à 3 h*
Prép N2 =: de Nq (binary)	An object Prép N2 can have de as its preposition and an abstract noun as its head noun Example : *Il élimine de son programme que le départ puisse être retardé*
Prép N2 =: Prép ce Qu P = Qu P (binary)	Preposition and ce before a sentential complement N2 in que can be erased Example : *Luc s'est renseigné auprès de Marie de ce qu'il est invité = Luc s'est renseigné auprès de Marie qu'il est invité*
Prép N2 =: Prép Qu P = Ppv (binary)	A sentential complement Prép N2 can pronominalize into a clitic Example : *On a autorisé Paul à faire ça = On l'y a autorisé ; Luc félicite Marie d'avoir fait ça = Luc l'en félicite*

TABLE E.37: Documentation des propriétés des tables de verbes (37)

Feature	Description of the feature
Prép N2 N0 V N1 (binary)	An object Prép N2 can appear at the beginning of the sentence Example : *Je tire une conclusion de cela = De cela je tire une conclusion*
Prép N2hum = Ppv (binary)	An object Prép N2, denoting a person, can pronominalize into a clitic Example : *Paul a crié à Luc de venir = Paul lui a crié de venir*
Prép N2hum = Ppv =: lui (binary)	An object Prép N2 with a human meaning can be pronominalized into the clitic pronoun *lui* Example : *Max passe un stylo à Ida = Max lui passe un stylo*
Prép N2-hum = Ppv (binary)	An object Prép N2, denoting a non-human entity, can pronominalize into a clitic Example : *Paul consacre son temps à l'écriture = Paul y consacre son temps*
Prép N2-hum = Ppv =: lui (binary)	An object Prép N2 with a non-human meaning can be pronominalized into the clitic pronoun *lui* Example : *Paul a consacré ce rayon à ses bouquins sur César = Paul leur a consacré ce rayon*
Prép N3 = Prép là (binary)	A sentential complement Prép N3 can be pronominalized into Prép là ; if the preposition is *à*, it is erased Example : (no Example known)
Prép N3 =: Prép Qu P = Ppv (binary)	Clitic pronominalization of a sentential complement Prép N3 ; the preposition can be empty Example : *Max a donné à Luc comme principe qu'il devait avouer = Max l'a donné à Luc comme principe*
Prép N3hum = Ppv (binary)	Clitic pronominalization of an object Prép N3 denoting a person ; the Prép can be empty Example : *Max a reçu de Luc comme otage un de ses enfants = Max l'a reçu de Luc comme otage*
Prép N3-hum = Ppv (binary)	Clitic pronominalization of an object Prép N3 denoting a non-human entity ; the Prép can be empty Example : *Paul a donné à Luc comme travail ce classement = Paul l'a donné à Luc comme travail*
Prép V0-inf W = Ppv (binary)	An infinitive clause taking the form Prép V0-inf W, with an implicit subject coreferent to the subject of the main clause, can pronominalize into a clitic ; the preposition can be empty Example : *Paul s'empresse de démentir = Paul s'en empresse ; Paul court chercher du pastis = Paul y court*
Prép0 =: de (binary)	The subject can become an object introduced by the preposition *de* Example : *Les dauphins abondent dans la baie = La baie abonde de dauphins*
Prép0 =: en (binary)	The subject can become an object introduced by the preposition *en* Example : *Les dauphins abondent dans la baie = La baie abonde en dauphins*
Prép1 (lexical)	Prepositions of the object N1 Example : *Max va jusqu'à exiger des dommages :* **jusqu'à** *; Le verre va tomber :* <**E**> *; Qu'Ida est idiote éclate aux yeux de tous :* **Loc**
Prép1 =: à (binary)	An object N1 can be introduced by the preposition *à* Example : *Paul se mesure* **à Luc**
Prép1 =: avec (binary)	An object N1 can be introduced by the preposition *avec* Example : *Paul se mesure* **avec Luc**
Prép1 =: contre (binary)	An object N1 can be introduced by the preposition *contre* Example : *Paul se bat* **contre Luc**

TABLE E.38: Documentation des propriétés des tables de verbes (38)

Feature	Description of the feature
Prép1 =: dans (binary)	An object N1 can be introduced by the preposition *dans* Example : *Le si dissone* **dans cet accord**
Prép1 =: d'avec (binary)	An object N1 can be introduced by the preposition *d'avec* Example : *L'avis de Luc diverge* **d'avec le consensus**
Prép1 =: de (binary)	An object N1 can be introduced by the preposition *de* Example : *L'avis de Luc diverge* **du consensus**
Prép2 (lexical)	Prepositions of the object N2 Example : *Max a accrédité auprès des parents la nouvelle que Luc est mort :* **auprès de** *; Max a encadré dans ce texte que Luc était absent :* **Loc** *; Max a pour preuve de cela qu'il ne s'est pas montré :* **pour+comme**
Prép2 =: à (binary)	An object N2 can be introduced by the preposition *à* Example : *Léa compare Luc* **à un singe** *; Paul joue sa voiture* **à Luc** *au poker*
Prép2 =: auprès de (binary)	An object N2 can be introduced by the preposition *auprès de* Example : *Max excuse Léa* **auprès de Luc** *de n'être pas venu*
Prép2 =: auprès de = à (binary)	The preposition *auprès de* introducing an object can be substituted by the preposition *à* Example : *Paul se vante* **(auprès de+à) Paul** *de ce qu'il a gagné*
Prép2 =: avec (binary)	An object N2 can be introduced by the preposition *avec* Example : *Le maire a marié Paul* **avec Léa** *; Luc aromatise sa soupe* **avec du pistou** *; Paul parie une bouteille* **avec Luc** *qu'Eva ne viendra pas*
Prép2 =: contre (binary)	An object N2 can be introduced by the preposition *contre* Example : *Luc protège sa voiture* **contre la pluie** *d'une bâche*
Prép2 =: dans (binary)	An object N2 can be introduced by the preposition *dans* Example : *Luc ceinture la ville* **dans des remparts**
Prép2 =: d'avec (binary)	An object N2 can be introduced by the preposition *d'avec* Example : *Le juge a divorcé Paul* **d'avec Léa**
Prép2 =: de (binary)	An object N2 can be introduced by the preposition *de* Example : *Luc protège sa voiture* **de la pluie** *d'une bâche ; Luc discerne le vert* **du rouge** *; Luc aromatise sa soupe* **de pistou**
Prép2 =: en (binary)	An object N2 can be introduced by the preposition *en* Example : *Paul a équipé ses soldats* **en armes légères** *; Luc rend cette expression* **en anglais** *dans son texte*
Prép2 =: pour (binary)	An object N2 can be introduced by the preposition *pour* Example : *Max donne ce texte à Luc* **pour un chef-d'oeuvre**
Prép3 (lexical)	Prepositions of the object N3 Example : *Max désigne Luc à Léa pour faire ce travail :* **pour** *; Max a reçu de Luc comme garantie qu'il aurait une prime :* **<E>**
Prép3 =: à (binary)	An object N3 can be introduced by the preposition *à* Example : *Paul joue sa voiture à Luc* **au poker**
Prép3 =: avec (binary)	An object N3 can be introduced by the preposition *avec* Example : *Max a protégé sa voiture de la pluie* **avec une bâche**
Prép3 =: comme (binary)	An object N3 can be introduced by the preposition *comme* Example : *Max donne à Luc ce cheval* **comme gagnant**
Prép3 =: contre (binary)	An object N3 can be introduced by the preposition *contre* Example : *La banque a changé à Max son argent* **contre des dollars**
Prép3 =: dans (binary)	An object N3 can be introduced by the preposition *dans* Example : *Paul a rendu cette expression en anglais* **dans son texte**

TABLE E.39: Documentation des propriétés des tables de verbes (39)

Feature	Description of the feature
Prép3 =: de (binary)	An object N3 can be introduced by the preposition *de* Example : *Max a costumé Luc en clown* **d'une combinaison rouge**
Prép3 =: en (binary)	An object N3 can be introduced by the preposition *en* Example : *Max a traduit ce texte d'anglais* **en turc**
Prép3 =: par (binary)	An object N3 can be introduced by the preposition *par* Example : *Max a protégé sa voiture de la pluie* **par une bâche**
Prép3 =: pour (binary)	An object N3 can be introduced by the preposition *pour* Example : *Max donne ce texte à Luc* **pour un chef-d'oeuvre**
Prép3 =: sur (binary)	An object N3 can be introduced by the preposition *sur* Example : *Max a réduit 30 F à Luc* **sur le prix de cet objet**
Qu N0 V (Adj) (binary)	From a sentential complement in the indicative mood taking the form que N être Adj W, the noun phrase N can be extracted during the formation of a relative clause ; the rest of the sentential complement is conserved in the form of Adj W Example : *J'ai des doutes sur une opération. L'intéressé certifie que cette opération est véridique = J'ai des doutes sur une opération que l'intéressé certifie véridique*
Qu N0 V (être Adj) (binary)	From a sentential complement in the indicative mood taking the form que N être Adj W, the noun phrase N can be extracted during the formation of a relative clause ; the rest of the sentential complement becomes an infinitive clause être Adj W Example : *J'ai des doutes sur une opération. L'intéressé certifie que cette opération est véridique = J'ai des doutes sur une opération que l'intéressé certifie être véridique*
Qu P = de V0-inf W (binary)	A direct sentential complement in the indicative mood, with a subject coreferent to the main subject, can be replaced by an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to the main subject Example : *Paul feint qu'il a un malaise = Paul feint d'avoir un malaise*
Qu P = V0-inf W (binary)	A sentential complement in the indicative mood, with a subject coreferent to the main subject, can be replaced by a non-prepositional infinitive clause with an implicit subject coreferent to the main subject Example : *Paul compte qu'il viendra = Paul compte venir*
Qu Psubj = de V0-inf W (binary)	A direct sentential complement in the subjunctive mood, with a subject coreferent to the main subject, can be replaced by an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to the main subject Example : *Paul propose à Marie qu'il la conduise = Paul propose à Marie de la conduire*
Qu Psubj = de V2-inf W (binary)	A direct sentential complement in the subjunctive mood, with a subject coreferent to an object N2, can be replaced by an infinitive clause introduced by the preposition *de*, with an implicit subject coreferent to N2 Example : *Paul conseille à Luc qu'il parte = Paul conseille à Luc de partir*

TABLE E.40: Documentation des propriétés des tables de verbes (40)

Feature	Description of the feature
Qu Psubj = V0-inf W (binary)	A (possibly theoretical) sentential complement in the subjunctive mood, with a subject coreferent to the main subject, can be replaced by a non-prepositional infinitive clause with an implicit subject coreferent to the main subject Example : **Paul désire qu'il boive = Paul désire boire*
Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W) (binary)	A type of subject raising : a sentential complement in the subjunctive mood taking the form que Ni Vsubj W can be replaced with a constituent Ni followed by an infinitive clause Vi-inf W introduced by the preposition *de* Example : *Paul empêche que Pierre vienne = Paul empêche Pierre de venir*
Sfx =: age (binary)	A deverbal noun related with V with the suffix *-age* is used in a sentence syntactically and semantically connected to the basic construction Example : *Luc a paillé la chaise d'un paillage rustique*
Sfx =: ment (binary)	A deverbal noun related with V with the suffix *-ment* is used in a sentence syntactically and semantically connected to the basic construction Example : *Luc a empierré le chemin d'un empierrement solide*
Sfx =: tion (binary)	A deverbal noun related with V with the suffix *-tion* is used in a sentence syntactically and semantically connected to the basic construction Example : *Luc a irrigué son champ d'une irrigation efficace*
Sfx =: ure (binary)	A deverbal noun related with V with the suffix *-ure* is used in a sentence syntactically and semantically connected to the basic construction Example : *Paul a épluché les patates de leur peau épaisse (la peau est une "épluchure"). Luc a doublé son manteau d'une doublure de soie.*
sur combien ? (binary)	An additional complement introduced by the preposition *sur* and containing a cardinal numeral as a determiner denotes a portion of the place involved, and answers a question with *sur combien ?* Example : *Luc a descendu le Nil sur 300 km ; Luc a acheminé les colis sur 1000 km*
Tc =: futur (binary)	The infinitive clause can contain a future adverb while the main verb is in the present tense Example : *Paul renonce à partir demain ; Il incombe à l'intéressé de s'inscrire à partir de demain ; Luc jubile de se lever tard demain*
Tc =: passé (binary)	The infinitive clause can be at a compound tense while the main verb is in the present tense Example : *Paul engueule Luc d'avoir fait ça ; Cela compte pour Luc d'avoir réussi ; Luc jubile d'avoir réussi*
Tc =: présent (binary)	The infinitive clause can contain a present adverb while the main verb is in the present tense Example : *Paul passe pour être encore un bon amant*
thèmeN1 (lexical)	Prototypical example of a noun which can be the head of the object N1 Example : *Luc applique cette loi ; Luc collabore avec l'ennemi*
Tp = Tc (binary)	A time adverb in the main clause cannot coexist with a semantically incompatible time adverb in the infinitive clause Example : **Paul manque à ce moment de faire une gaffe demain ; *Luc passe en ce moment boire un coup ce soir ; *Luc traîne Marie maintenant voir un film tout à l'heure*

Table E.41: Documentation des propriétés des tables de verbes (41)

Feature	Description of the feature
trajet (binary)	Additional locative complement introduced by the preposition *sur* or *le long de* and interpreted as a place of passage ; the sentence may denote either a movement through the passage, or a stative situation where a person or thing can travel through the passage Example : *L'eau dégouline sur le mur ; cf. *Luc parvient de la fenêtre au balcon sur la corniche ; Le rocher avance dans la mer le long de la falaise*
V = convertir en V-n (binary)	Paraphrase of the form N0 convertir/transformer N1 en V-n or N0 faire de N1 V-n, where V-n is a deverbal noun related to V Example : *Le temps a parcheminé ce manuscrit = Le temps a converti/transformé ce manuscrit en parchemin ; Paul a déifié Marie = Paul a fait de Marie une déesse*
V = convertir en V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphrase of the form N0 convertir/transformer N1 en V-n or N0 faire de N1 V-n Example : *parchemin*
V = mettre en V-n (binary)	Paraphrase of the form N0 mettre N1 en V-n, where V-n is a deverbal noun related to V Example : *Paul a bobiné le fil = Paul a mis le fil en bobine*
V = mettre en V-n (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphrase of the form N0 mettre N1 en V-n Example : *bobine*
V mouvement (binary)	The sentence denotes a movement Example : *Luc percute le mur ; Luc croise Léa*
V statique (binary)	The sentence denotes a stative situation Example : *Luc habite cette maison ; L'avenue croise le boulevard*
V1-inf W = Ppv (binary)	An infinitive clause taking the form V1-inf W, with an implicit subject coreferent to an object N1, can pronominalize into a clitic Example : *Paul emmène Marie faire des courses = Paul y emmène Marie*
V-adj (lexical)	Form of the deverbal adjective related to V in the sentence N0 rendre N1 V-adj, or N0 rendre N1 plus V-adj, or N1 être V-adj à N2 *On a libéré Paul (E+de ses liens) = On a rendu Paul libre (E+de ses liens) ; On a allégé le paquet de 100 g = Le paquet est plus léger de 100 g ; On a imperméabilisé ce tissu à l'eau = Ce tissu est imperméable à l'eau*
Vc =: aimer (binary)	The infinitive clause can contain the verb *aimer* Example : *Cela tombe mal pour Roméo d'aimer Juliette ; Luc en vient à aimer Marie*
Vc =: avoir (binary)	The infinitive clause can contain the verb *avoir* Example : *Il en coûte à Luc d'avoir cette voiture ; Luc jubile d'avoir une voiture*
Vc =: devoir (binary)	The infinitive clause can contain the verb *devoir* Example : *Paul rechigne à devoir faire ça ; Il arrive à Luc de devoir consulter son agenda ; cf. *Paul demande à devoir faire ça*
Vc =: être (binary)	The infinitive clause can contain the verb *être* Example : *Il en coûte à Luc d'être ici ; Luc jubile d'être ici*
Vc =: pouvoir (binary)	The infinitive clause can contain the verb *pouvoir* Example : *Il arrive à Luc de pouvoir se lever ; Paul aspire à pouvoir faire ça*

Table E.42: Documentation des propriétés des tables de verbes (42)

Feature	Description of the feature
Vc =: savoir (binary)	The infinitive clause can contain the verb *savoir* Example : *Il manque à Luc de savoir faire rire ; Paul aspire à savoir faire ça*
Vc =: vouloir (binary)	The infinitive clause can contain the verb *vouloir* Example : *Il arrive à Luc de vouloir déménager ; Ceci revient pour Luc à vouloir déménager*
V-n attache (binary)	Paraphase of the form N0 attacher N1 à N2 avec V-n, where V-n is a deverbal noun related to V and denoting a tying system Example : *On a agrafé la feuille bleue sur la feuille rouge = On a attaché la feuille bleue à la feuille rouge avec une agrafe*
V-n attache (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphase of the form N0 attacher N1 à N2 avec V-n, denoting a tying system Example : *agrafe*
V-n chemin (binary)	Semantically related sentence of the form N1 passer par V-n, where V-n is a deverbal noun related to V and denoting a path or a place of passage Example : *Luc canalise l'eau = L'au passe par un canal ; Luc draine le liquide = Le liquide passe par un drain*
V-n chemin (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a semantically related sentence of the form N1 passer par V-n, denoting a path or a place of passage Example : *canal*
V-n instrument (binary)	Existence of a deverbal noun related to V and denoting an instrument Example : *Luc balaye la poussière/balai ; Luc canonne le bastion/canon ; Luc tamise la farine/tamis*
V-n instrument (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V denoting an instrument Example : *balai*
V-n résultat (binary)	Semantically related sentence of the form N0 faire V-n, where V-n is a deverbal noun related to V and denoting the result Example : *On a ligaturé les deux lèvres de la plaie = On a fait une ligature*
V-n résultat (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a semantically related sentence of the form N0 faire V-n, denoting the result Example : *ligature*
V-n transport (binary)	Paraphase of the form N0 transporter N1 (dans+avec) V-n, where V-n is a deverbal noun related to V and denoting a transport mode Example : *Luc véhicule Marie = Luc transporte Marie (dans+avec) un véhicule ; Luc brouette le foin = Luc transporte le foin (dans+avec) une brouette*
V-n transport (forme V-n) (lexical)	Form of a deverbal noun morphologically associated with V, used in a paraphase of the form N0 transporter N1 (dans+avec) V-n, denoting a transport mode Example : *véhicule*

Table E.43: Documentation des propriétés des tables de verbes (43)



Formules définitoires des tables de verbes distributionnels

Les Tab. F.1 à Tab. F.20 représentent la définition formelle de chaque table de verbes qui justifie leur classification. Les tables, accompagnées d'un exemple, sont classées selon un ordre de priorité à appliquer afin de savoir dans quelle table est incluse (ou doit être ajoutée) une entrée. Afin de pouvoir appliquer plus facilement cet ordre de priorité, on peut se référer à l'arbre de classement des verbes, inclus dans l'annexe G. La première distinction est le nombre d'arguments de la construction de base qui peut être compris entre un et quatre. Chaque table est donc associée à sa *formule définitoire*, composée d'un ensemble de disjonctions, conjonctions et négations de propriétés syntaxico-sémantiques, autrement dit, un ensemble de propriétés séparées par des ou logiques, des et logiques et des négations non. Cette formule contient tout d'abord la construction de base (première ligne), ainsi que les autres propriétés définitoires, chacune séparées par des et logiques. Elle y inclut aussi bien les propriétés définitoires vraies que fausses (introduites par la négation non) pour l'ensemble d'une table, ainsi que les disjonctions de propriétés codées dans les tables. Ces dernières sont représentées par un ensemble de propriétés séparées par des ou logiques, qui signifient qu'au moins une des propriétés parmi l'ensemble est vraie. Un exemple détaillé (non complet) est donné en 5.2.2.

Quatre arguments :
(N0 V N1 Loc N2 V1-inf W) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N3 =: Qu Pind) et non (N3 =: Qu Psubj) → **3** (*Paul envoie Luc au village faire le marché*)
(N0 V Prép N1 Prép N2 Prép N3) et ((N3 =: Qu Pind) ou (N3 =: Qu Psubj)) → **18** (*Ça a servi à Paul de prétexte pour ne pas venir ; Paul prétexte une maladie auprès de Luc pour ne pas venir*)
(N0 V N1 Loc N2 source Loc N3 destination) et (N2 lieu source de N1) et (N3 lieu de destination de N1) et non (N1 =: N-hum) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (Prép3 =: <E>) et non (N3 =: Qu Pind) et non (N3 =: Qu Psubj) → **38LH** (*On a refoulé Luc de France dans son pays*)
(N0 V N1 Loc N2 source Loc N3 destination) et (N1 =: N-hum) et (N2 lieu source de N1) et (N3 lieu de destination de N1) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (Prép3 =: <E>) et non (N3 =: Qu Pind) et non (N3 =: Qu Psubj) → **38L** (*Luc transvase le vin de la cuve dans le tonneau*)

TABLE F.1: Formules définitoires dans les tables de verbes (1)

(N0 V N1 Prép N2 Prép N3) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (Prép3 =: <E>) et non (N3 =: Qu Pind) et non (N3 =: Qu Psubj) et (non (N2 lieu source de N1) ou non (N3 lieu de destination de N1)) → **38RR** (*Luc protège le livre contre la pluie d'un plastique*)
Trois arguments :
(N0 V Loc N1 V0-inf W) et non (Prép1 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: si P ou si P) → **2** (*Paul court à la pharmacie chercher de l'aspirine*)
(N0 V N1 à N2) et (N2 =: Nhum) et non (Prép2 # à) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 =: si P ou si P)) → **9** (*Paul crie à Léa que tout est fini*)
(N0 V N1 à N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (Prép2 # à) et ((N2 =: Qu Pind) ou (N2 =: Qu Psubj) ou (N2 =: V0-inf W) ou (N2 =: V1-inf W)) → **11** (*Paul dresse Léa à tout ranger*)

Table F.2: Formules définitoires dans les tables de verbes (2)

(N0 V N1 Prép N2) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj)) et ((Prép2 # à) ou non (N2 =: Nhum)) → **10** (*Ceci comporte pour Flo qu'elle doit partir tôt*)
(N0 V N1 de N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (Prép2 # de) et ((N2 =: Qu Pind) ou (N2 =: Qu Psubj) ou (N2 =: V1-inf W)) → **13** (*Paul informe Luc de ce que Léa arrive*)
(N0 V N1 Prép N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (Prép2 # à+de) et non (N2 lieu source de N1) et non (N2 lieu de destination de N1) et non (N2 lieu statique de N1) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N0 V N1 et N2) et ((N2 bénéficiaire) ou (N2 détrimentaire)) → **36DT** (*Paul offre des fleurs à Léa ; Ida a volé une fleur à Luc ; Ida a reçu une fleur de Luc*)

Table F.3: Formules définitoires dans les tables de verbes (3)

(N0 V N1 Prép N2) et (N0 V N1 et N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et ((N2 lieu source de N1) ou (N2 lieu de destination de N1)) → **36SL** (*Luc colle la fiche bleue sur la fiche jaune*)
(N0 V N1 Prép N2) et (N0 V N1 et N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (Prép2 =: <E>) et non (N2 lieu source de N1) et non (N2 lieu de destination de N1) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) → **36S** (*Luc compare Guy (à+avec+et) Louis*)

Table F.4: Formules définitoires dans les tables de verbes (4)

(N0 V N1 à N2) et non (N0 V N1 et N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (N2 lieu source de N1) et non (N2 lieu de destination de N1) et non (N2 lieu statique de N1) et non (Prép2 # à) et non (N2 bénéficiaire) et non (N2 détrimentaire) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) → **36R** (*Luc conjugue le verbe au futur*)
(N0 V N1 de N2) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (Prép2 =: à+<E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 lieu source de N0) et non (N2 lieu source de N1) et non (N2 lieu de destination de N1) et non (N2 lieu statique de N1) et non (N2 bénéficiaire) et non (N2 détrimentaire) et non (N0 V N1 et N2) et ((N1 lieu source de N2) ou (N1 détrimentaire)) → **37E** (*Luc débarrasse le salon de tous les meubles*)

Table F.5: Formules définitoires dans les tables de verbes (5)

(N0 V N1 Prép N2)
et (Prép2 =: de)
et (N1 bénéficiaire)
et non (N1 =: Qu Pind)
et non (N1 =: Qu Psubj)
et non (N1 =: V0-inf W)
et non (Prép2 =: à+<E>)
et non (N2 =: Qu Pind)
et non (N2 =: Qu Psubj)
et non (N2 =: V1-inf W)
et non (N2 lieu source de N0)
et non (N2 lieu source de N1)
et non (N2 lieu de destination de N1)
et non (N2 lieu statique de N1)
et non (N2 bénéficiaire)
et non (N2 détrimentaire)
et non (N0 V N1 et N2)
→ **37M1** (*L'héritage a nanti Luc d'une fortune*)

(N0 V N1 Prép N2)
et (Prép2 =: de)
et (N1 lieu de destination de N2)
et non (N1 =: Qu Pind)
et non (N1 =: Qu Psubj)
et non (N1 =: V0-inf W)
et non (Prép2 =: à+<E>)
et non (N2 =: Qu Pind)
et non (N2 =: Qu Psubj)
et non (N2 =: V1-inf W)
et non (N2 lieu source de N0)
et non (N2 bénéficiaire)
et non (N2 détrimentaire)
et non (N0 V N1 et N2)
→ **37M2** (*Luc a hachuré la feuille de traits rouges*)
→ **37M3** (*Luc a labouré le terrain de profonds sillons*)
→ **37M4** (*Luc a jonché le sol de papiers gras*)
→ **37M5** (*On a goudronné la route d'une épaisse couche de goudron*)
→ **37M6** (*On a gainé le fil d'une gaine de plastique rouge*)

TABLE F.6: Formules définitoires dans les tables de verbes (6)

(N0 V N1 Loc N2 destination) et (N2 lieu de destination de N1) et non (N1 =: N-hum) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **38LHD** (*On a relégué Max outre-mer*)
(N0 V N1 Loc N2 source) et (N2 lieu source de N1) et non (N1 =: N-hum) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **38LHS** (*Le choc a désarçonné Max de son cheval*)
(N0 V N1 Loc N2) et (N2 lieu statique de N1) et non (N1 =: N-hum) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **38LHR** (*Max héberge Luc dans son studio*)

Table F.7: Formules définitoires dans les tables de verbes (7)

(N0 V N1 Loc N2 destination) et (N1 =: N-hum) et (N2 lieu de destination de N1) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **38LD** (*Luc a déposé le paquet sur le lit*)
(N0 V N1 Loc N2 source) et (N1 =: N-hum) et (N2 lieu source de N1) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **38LS** (*Luc a ôté le livre du lit*)
(N0 V N1 Loc N2) et (N1 =: N-hum) et (N2 lieu statique de N1) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **38LR** (*Luc conserve son vin dans le grenier*)

Table F.8: Formules définitoires dans les tables de verbes (8)

(N0 V N1 en N2) et (N2 être V-n) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N1 =: si P ou si P) et non (N1 lieu de destination de N0) et non (Prép2 # en) et non (N2 lieu source de N1) et non (N2 lieu de destination de N1) et non (N2 lieu statique de N1) et non (N2 =: Npl obl) et non (N0 V N1 et N2) → **32CV** (*Ce procédé caramélise le sucre en une pâte aromatique*)
(N0 V N1 en N2) et (N2 =: Npl obl) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N2) et non (N1 lieu source de N2) et non (N1 bénéficiaire) et non (N1 détrimentaire) et non (Prép2 # en+entre) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N0 V N1 et N2) → **38PL** (*Luc a coupé le gâteau en trois parts*)

Table F.9: Formules définitoires dans les tables de verbes (9)

(N0 V N1 Prép N2) et (Prép2 # à) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) et non (N1 lieu de destination de N0) et non (N1 lieu de destination de N2) et non (N1 lieu source de N2) et non (N1 bénéficiaire) et non (N1 détrimentaire) et non (N2 lieu source de N1) et non (N2 lieu de destination de N1) et non (N2 lieu statique de N1) et non (Prép2 =: <E>) et non (N2 bénéficiaire) et non (N2 détrimentaire) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V0-inf W) et non (N2 =: V1-inf W) et non (N0 V N1 et N2) et (non (Prép2 =: en) ou (non (N2 =: Npl obl) et non (N2 être V-n))) → **38R** (*Luc a prévenu Guy contre ce genre de type*)
(N0 V Loc N1 V0-inf W) et (Prép1 =: <E>) → **2T** (*Paul dévale l'escalier chercher Marie*)
(N0 V N1 Prép N2) et (Prép2 =: <E>) et non (N1 =: V0-inf W) et non (Prép2 # <E>+comme) et non (N2 =: V0-inf W) → **39** (*On a élu Guy président*)
(N0 V à N1 Prép N2) et non (Prép1 # à) et non (Prép2 # avec+pour) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V1-inf W) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj)) → **14** (*Cette règle équivaut pour Paul à ce que Léa soit élue*)

TABLE F.10: Formules définitoires dans les tables de verbes (10)

(N0 V de N1 Prép N2) et non (Prép1 # de) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V1-inf W) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj)) → **15** (*Paul répond devant le juge de ce que Léa a fauté*)
(N0 V Prép N1 Prép N2) et non (Prép2 =: <E>) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (non (N1 =: V0-inf W) et ((Prép1 # <E>) ou (non (Prép2 =: à) et non (Prép2 =: de))) et ((Prép1 # avec+pour) ou non (Prép2 =: à)))) et ((N2 =: Qu Pind) ou (N2 =: Qu Psubj) ou (N2 =: V1-inf W)) → **16** (*Flo déduit que Guy est arrivé de ce qu'il y a du désordre ; Flo engage Guy pour danser ; Flo apprend à Guy à danser*)
(N0 V Loc N1 source Loc N2 destination) et (N1 lieu source de N0) et (N2 lieu de destination de N0) et non (Prép1 =: <E>) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N1 =: V2-inf W) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: si P ou si P) et non (N0 V N1 et N2) → **35L** (*Luc passe du salon à la cuisine*)
(N0 V Prép N1 Prép N2) et non (Prép1 =: <E>) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V2-inf W) et non (Prép2 =: <E>) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V1-inf W) et (non (N1 lieu source de N0) ou non (N2 lieu de destination de N0)) → **35RR** (*Léa change de voiture avec Luc*)

Table F.11: Formules définitoires dans les tables de verbes (11)

Deux arguments :
(N0 V N1) et (N0 =: Qu P) et (N1 =: Nhum) et non (N1 =: Nconc) et non (N1 =: Npc) et non (N1 =: V0-inf W) et non (Qu Psubj =: Qu Ni Vsubj W = (Ni) (de V-inf W)) et non [extrap] et ((N0 =: V1-inf W) ou (non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P))) → **4** (*Que Paul vienne amuse Luc ; Que les enfants soient ensemble neutralise qu'ils soient agités*)
(N0 V N1) et non (N0 =: V1-inf W) et non (Qu Psubj =: Qu Ni Vsubj W = (Ni) (de V-inf W)) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 =: si P ou si P) ou (N1 =: combien P)) → **6** (*Paul estime que Luc a raison ; Luc contemple combien le ciel est beau*)
(N0 V N1) et (N1 =: Qu Psubj) et (Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W)) → **12** (*Paul apprécie (que Léa l'ait aidé+Léa de l'avoir aidé*)
(N0 V N1) et (thèmeN1) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 lieu de passage de N0) → **32R3** (*Luc a brisé ses chaussures ; Cet appareil adoucit l'eau*)

TABLE F.12: Formules définitoires dans les tables de verbes (12)

(N0 V N1) et [passif par] et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 lieu de passage de N0) et ((N0 rendre N1 V-adj) ou (N0 rendre N1 plus V-adj)) et ((N1 =: Nconc) ou non (N0 =: Qu P) ou non (N1 =: Nhum)) → **32RA** (*Ce rideau assombrit la pièce*)
(N0 V N1) et (N1 apparition) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) → **32A** (*Luc construit une cabane*)
(N0 V N1) et (N1 disparition) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) → **32D** (*L'explosion a soufflé le bar*)

Table F.13: Formules définitoires dans les tables de verbes (13)

(N0 V N1) et (N1 =: Npl obl) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) et non (N1 apparition) et non (N1 disparition) → **32PL** (*Luc entasse les cubes*)
(N0 V N1) et (N1 =: Nhum) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) et non (N1 =: N-hum) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj) → **32H** (*Luc a violé Ida*)
(N0 V N1) et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non [passif par] et non [passif de] et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 lieu de passage de N0) → **32NM** (*Le sac pèse 10 kg*)

TABLE F.14: Formules définitoires dans les tables de verbes (14)

(N0 V N1) et (N1 =: N-hum) et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et ((N0 lieu source de N1) ou (N0 lieu de destination de N1) ou (N0 lieu statique de N1)) → **38L0** (*Luc a inhalé le gaz*)
(N0 V N1) et (N1 =: N-hum) et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et ((N1 lieu source de N0) ou (N1 lieu de destination de N0) ou (N1 lieu statique de N0) ou (N1 lieu de passage de N0)) et ((N1 être Vpp W) ou non (N0 V N1 Loc N1pc W) ou non (N1 =: Npc) ou non [passif par]) → **38L1** (*Luc gagne la porte*)

TABLE F.15: Formules définitoires dans les tables de verbes (15)

(N0 V N1) et (N0 V N1 Loc N1pc W) et (N1 =: Npc) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (N1 =: Nabs) et non (thèmeN1) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj) et non (N1 être Vpp W) → **32CL** (*Luc embrasse (le front de Léa+Léa sur le front)*)
(N0 V N1) et (N1 =: N-hum) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (N1 =: Nabs) et non (thèmeN1) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 lieu de passage de N0) et non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj) et ((N1 être Vpp W) ou non (N0 V N1 Loc N1pc W) ou non (N1 =: Npc)) → **32C** (*Luc a abîmé le livre*)

Table F.16: Formules définitoires dans les tables de verbes (16)

(N0 V N1) et (N1 =: Nabs) et (N0 V (N1 de N1c) = N0 V (N1c) (Prép N1)) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 lieu de passage de N0) et non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj) → **32R1** (*Luc singe (les attitudes de Léa+Léa dans ses attitudes)*)
(N0 V N1) et (N1 =: Nabs) et [passif par] et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) et non (N1 =: V0-inf W) et non (thèmeN1) et non (N1 apparition) et non (N1 disparition) et non (N1 =: Npl obl) et non (N0 lieu source de N1) et non (N0 lieu de destination de N1) et non (N0 lieu statique de N1) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 lieu de passage de N0) et non (N0 V (N1 de N1c) = N0 V (N1c) (Prép N1)) et non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj) → **32R2** (*Luc brandit le livre ; Luc copie un texte*)

Table F.17: Formules définitoires dans les tables de verbes (17)

(N0 V Prép V0-inf W) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: combien P) → **1** (*Paul cesse de travailler ; Paul cesse le travail ; *Paul cesse que Marie travaille*)
(N0 V Prép N1) et (N0 =: Qu P) et [extrap] et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj) → **5** (*Que Marie vienne importe à Paul ; Il importe à Paul que Marie vienne*)
(N0 V à N1) et non (Prép1 # à) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj)) → **7** (*Paul consent à ce que Marie fasse ça*)
(N0 V de N1) et non (Prép1 # de) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj)) → **8** (*Paul s'insurge de ce que Marie ait fait ça*)
(N0 V à N1) et non (Prép1 # à) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N1 = où) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et (non (N0 =: Qu P) ou non [extrap]) → **33** (*Paul compatit à mes problèmes*)
(N0 V Prép N1) et (Prép1 =: avec) et (N0 et N1 V) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et (non (N0 =: Qu P) ou non [extrap]) → **35S** (*Paul flirte avec Marie ; Paul et Marie flirtent*)

TABLE F.18: Formules définitoires dans les tables de verbes (18)

(N0 V Prép N1) et (Prép1 # à+<E>) et non (N1 = où) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N0 et N1 V) et (non (N0 =: Qu P) ou non [extrap]) et ((non (N1 =: Qu Pind) et non (N1 =: Qu Psubj)) ou ((Prép1 # à) et (Prép1 # de))) et ((N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou non (N1 =: V0-inf W)) → **35R** (*Ça a dégénéré en bagarre ; Paul dîne d'une salade*)
(N0 V Loc N1 source) et (N1 lieu source de N0) et non (N0 =: Qu P) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N0 et N1 V) et non (N1 V de N0) → **35LS** (*Ce vin provient de Carnas*)
(N0 V Loc N1 destination) et (N1 lieu de destination de N0) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N0 et N1 V) et (non (N0 =: Qu P) ou non [extrap]) → **35LD** (*Le bateau accoste au quai*)
(N0 V Loc N1) et (N1 V de N0) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N0 et N1 V) et (non (N0 =: Qu P) ou non [extrap]) et ((N1 lieu source de N0) ou (N1 lieu statique de N0)) → **34L0** (*Les abeilles grouillent dans le jardin = Le jardin grouille d'abeilles*)

TABLE F.19: Formules définitoires dans les tables de verbes (19)

(N0 V Loc N1) et (N1 lieu statique de N0) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N0 et N1 V) et non (N1 V de N0) et (non (N0 =: Qu P) ou non [extrap]) → **35ST** (*Le pieu sort de l'eau ; Le chemin aboutit chez moi*)
(N0 V Loc N1) et (N1 = où) et non (N1 lieu source de N0) et non (N1 lieu de destination de N0) et non (N1 lieu statique de N0) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W) et non (N0 et N1 V) et non (N1 V de N0) et (non (N0 =: Qu P) ou non [extrap]) → **35LR** (*Paul farfouille dans le sac*)
Un argument :
(N0 V) et (N0 =: Nhum) et non (N0 =: N-hum) → **31H** (*Paul patiente*)
(N0 V) et (N0 =: N-hum) → **31R** (*(Paul+la terre) tremble ; Un accident s'est produit*)

TABLE F.20: Formules définitoires dans les tables de verbes (20)



Arbre de classement des verbes distributionnels

G.1 Conventions de lecture de l'arbre

G.1.1 Objectif

L'arbre de classement (Fig. G.1 à Fig. G.7) sert à déterminer à quelle classe appartient une entrée verbale donnée. Il est conçu pour un utilisateur qui connaît les propriétés de l'entrée et qui applique successivement les critères indiqués dans l'arbre.

G.1.2 Constructions prises en compte

Presque tous les critères concernent la construction qui comporte le sujet et tous les compléments essentiels, et que nous appelons construction de base (cf. 5.2.2). Par exemple, pour *transvaser*, la construction de base est illustrée par la phrase *Luc transvase le vin de la cuve dans le tonneau*, qui a un complément direct et deux compléments prépositionnels. Quelques critères concernent des variantes de la construction de base, par exemple *On élit Luc comme président / On élit Luc président*, et d'autres encore des constructions totalement différentes dans laquelle les arguments occupent d'autres positions, comme *Le vin est transvasé par Luc de la cuve dans le tonneau.*

G.1.3 Détermination de la construction de base

Avant d'utiliser l'arbre de classement sur une entrée, l'utilisateur détermine sa construction de base, car c'est celle-ci qui servira de référence pour l'application de

la plupart des critères. La construction de base est choisie parmi les différentes constructions qui relèvent de l'entrée, et dans lesquelles le verbe conserve le même sens. Ce choix peut être délicat et même en partie arbitraire, mais il repose sur des priorités :

– priorité à la construction qui comporte le plus de compléments essentiels, par exemple *Luc conjugue le verbe au futur* par rapport à *Luc conjugue le verbe* ;
– priorité à l'actif par rapport au passif, même lorsqu'il est moins employé, par exemple *Le paysage éberlue Max* par rapport à *Max est éberlué par le paysage* ;
– priorité à la construction qui comporte une préposition par rapport à celle sans préposition, par exemple *On élit Luc comme président* par rapport à *On élit Luc président* ;
– priorité à la complétive sur l'infinitive, par exemple *Luc craint qu'il ne pleuve* par rapport à *Luc craint d'être mouillé*, et donc complément direct ;
– lorsqu'une construction contient deux compléments essentiels dont l'un dénote un lieu, et situe par rapport à ce lieu une entité dénotée par l'autre complément, priorité à la construction dans laquelle le complément de lieu est prépositionnel, par exemple *Luc peint un portrait sur la cloison* par rapport à *Luc peint la cloison d'un portrait* ;
– lorsqu'il existe un complément direct et un complément avec une des prépositions *en* ou *entre*, la construction dans laquelle apparaît ce dernier a priorité par rapport aux éventuelles autres constructions qui comportent un autre complément prépositionnel, par exemple *Luc ventile le courrier en quatre tas* par rapport à *Luc ventile le courrier dans les services.*

G.1.4 Numérotation des arguments

Presque tous les critères font référence à un des arguments syntaxiques de la construction de base à travers une numérotation. Les arguments syntaxiques sont le sujet et les compléments essentiels. Ils sont supposés être numérotés à partir de 0 et conformément à l'ordre (ou à un des ordres possibles) des compléments dans la construction. Le sujet porte donc le numéro 0. De plus, on place les compléments directs avant les compléments indirects lorsque cet ordre est acceptable : *Luc formule ses réflexions à Marie* plutôt que *Luc formule à Marie ses réflexions.* Ces principes laissent parfois le choix entre plusieurs numérotations. Dans ce cas, l'arbre de classement tente de prévoir toutes les numérotations recevables, quitte à indiquer à la fin qu'il faut inverser deux numéros. Les prépositions éventuelles introduisant des compléments essentiels sont indicées par le numéro correspondant. Les groupes nominaux ou propositions constituant les arguments syntaxiques sont symbolisés par N indicé par le numéro. Ainsi, dans *Luc formule ses réflexions à Marie*, le symbole N1 représente *ses réflexions*, Prép2 symbolise *à* et N2 symbolise *Marie.*

G.1.5 Critères

Les critères utilisés dans l'arbre correspondent à des propriétés syntaxiques et sémantiques de l'entrée à classer. La plupart prennent la forme d'un des intitulés décrits dans la documentation des propriétés (cf. annexe E), par exemple

N1 =: Qu Pind qui indique la possibilité d'une complétive objet à l'indicatif en position N1. D'autres sont des formules booléennes sur de tels intitulés (cf. 5.2.2) : par exemple, (N1 =: Qu Pind) ou (N1 =: Qu Psubj) indique la possibilité d'une complétive objet à l'indicatif ou au subjonctif. Les quelques intitulés qui ne sont pas explicitement documentés utilisent les mêmes notations que les autres. Il faut appliquer successivement entre 2 et 14 critères pour déterminer à quelle classe appartient une entrée. Chaque étape propose un choix entre plusieurs critères exclusifs les uns des autres. Lorsqu'il y en a deux, ils sont souvent la négation logique l'un de l'autre. Par exemple, le critère associé à (N1 =: Qu Pind) ou (N1 =: Qu Psubj) est sa négation non (N1 =: Qu Pind) et non (N1 =: Qu Psubj).

G.1.6 Critères distributionnels

Beaucoup de critères contiennent le symbole =: et indiquent une valeur que peut prendre un des éléments de la construction de base (ou parfois plusieurs valeurs). Ainsi, N0 =: N-hum indique que le sujet N0 peut prendre comme valeur un groupe nominal N-hum dénotant une entité non humaine. Un tel critère n'indique pas une valeur exclusive : si le sujet peut aussi prendre d'autres valeurs, cela n'empêche pas que le critère donne un résultat positif. La seule exception à cette convention est le symbole Npl obl qui désigne un groupe nominal obligatoirement pluriel ou à sens collectif. Le symbole <E> représente l'absence de forme explicite, par exemple l'absence de préposition. Le symbole # se lit «différent de» et précède une ou plusieurs valeurs que l'élément **peut** ne pas prendre. Ainsi, Prép2 # à indique que la préposition peut avoir une valeur autre que *à*. Le critère Prép2 # <E> indique que la préposition peut avoir une forme explicite. Le critère non (Prép2 # <E>) indique qu'aucune préposition explicite ne peut apparaître en cette position.

G.1.7 Variantes de formulation

Un même critère peut généralement être formulé de plusieurs façons (cf. 5.2.2) : par exemple, dans le contexte des verbes transitifs directs à deux arguments, la possibilité d'une complétive objet à l'indicatif peut être notée N1 =: Qu Pind ou N0 V Qu Pind. La formulation choisie dans l'arbre de classement n'est pas toujours la même que celle choisie dans les intitulés des propriétés décrites dans les tables.

G.1.8 Résultat du classement

À l'issue de l'application des critères, l'arbre indique la classe[1] à laquelle appartient l'entrée. Dans certains cas, il indique en outre une ou plusieurs propriétés additionnelles que doit posséder l'entrée et qui n'ont pas été vérifiées dans les critères qui ont mené à cette branche. Par exemple, la branche menant à la classe 39 passe uniquement par 4 nœuds de l'arbre :

1. La notation **_part** indique qu'il existe plusieurs chemins menant à cette même classe.

3 arguments → Prép1 =: <E> → Prép2 =: <E> → non (N2 =: V0-inf W)
Mais elle indique comme propriétés additionnelles :
non (N1 =: V0-inf W) et non (Prép2 # <E>+comme)
En d'autres termes, les auteurs du Lexique-Grammaire ont constaté lors de leur travail que toutes les entrées qui vérifient à la fois les 4 propriétés ci-dessus vérifient également ces propriétés additionnelles.

G.2 Arbre de classement des verbes

4 arguments
- (N3 =: Qu Pind) ou (N3 =: Qu Psubj) **18**
  - non (N3 =: Qu Pind) et non (N3 =: Qu Psubj) et non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (Prép1 # <E>)
    - Prép3 =: <E>
      - non (Prép1 # <E>) et (N3 =: V1-inf W) **3**
  - non (Prép3 =: <E>) et non (Prép2 =: <E>)
    - (N2 lieu source de N1) et (N3 lieu de destination de N1)
      - N1 =: N-hum **38L**
      - non (N1 =: N-hum) **38LH**
    - non (N2 lieu source de N1) ou non (N3 lieu de destination de N1) ... **38RR**

FIGURE G.1: Arbre de classement des verbes (1)

3 arguments

- Prép1 =: <E>
  - Prép2 =: <E>
    - N2 =: V0-inf W **2T**
    - non (N2 =: V0-inf W)
      - non (N1 =: V0-inf W) et non (Prép2 # <E>+comme) **39**
  - non (Prép2 =: <E>)
    - (N2 =: Qu Pind) ou (N2 =: Qu Psubj) ou (N2 =: V1-inf W) ou (N2 =: V0-inf W)
      - (N1 =: Qu Pind) ou (N1 =: Qu Psubj) .... **16_part**
      - non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V0-inf W)
        - (Prép2 # à) et (Prép2 # de) .... **16_part**
        - non (Prép2 # à) **11**
        - non (Prép2 # de) **13**
    - non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V1-inf W) et non (N2 =: V0-inf W)
      - (N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 =: si P ou si P)
        - Prép2 # à .... **10_part**
        - non (Prép2 # à)
          - N2 =: Nhum **9**
          - non (N2 =: Nhum) .... **10_part**
      - non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: si P ou si P) et non (N1 =: V0-inf W) .... **voir Fig. G.3**
- non(Prép1 =: <E>)
  - (N2 =: Qu Pind) ou (N2 =: Qu Psubj) ou (N2 =: V1-inf W)
    - (N1 =: Qu Pind) ou (N1 =: Qu Psubj)
      - non (Prép2 =: <E>) .... **16_part**
    - non (N1 =: Qu Pind) et non (N1 =: Qu Psubj)
      - (Prép1 # avec+pour) ou (Prép2 # à)
        - non (Prép2 =: <E>) .... **16_part**
      - non (Prép1 # avec+pour) et non (Prép2 # à) .... **14_part**
  - non (N2 =: Qu Pind) et non (N2 =: Qu Psubj) et non (N2 =: V1-inf W) et non (N2 =: si P ou si P)
    - (N1 =: Qu Pind) ou (N1 =: Qu Psubj)
      - non (Prép1 # à)
        - non (Prép2 # avec+pour) .... **14_part**
      - non (Prép1 # de)
        - non (Prép2 =: <E>) **15**
    - non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 =: V2-inf W)
      - (N2 =: V0-inf W)
        - non (Prép2 # <E>) **2**
      - non (N2 =: V0-inf W)
        - (N1 lieu source de N0) et (N2 lieu de destination de N0)
          - non (N1 =: V0-inf W) et non (N0 V N1 et N2) **35L**
        - non (N1 lieu source de N0) ou non (N2 lieu de destination de N0)
          - non (Prép2 =: <E>) **35RR**

FIGURE G.2: Arbre de classement des verbes (2)

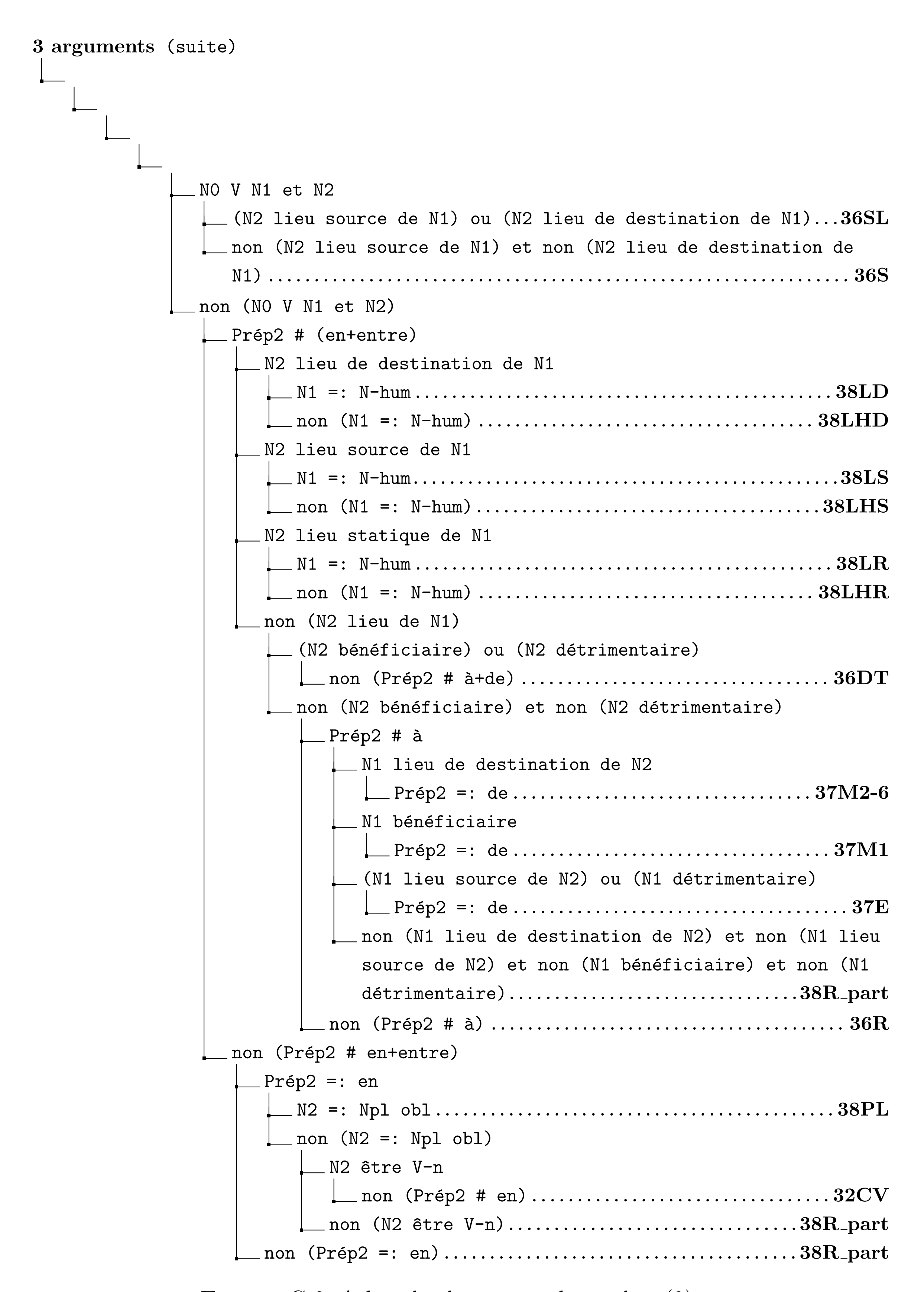


Figure G.3: Arbre de classement des verbes (3)

2 arguments

- Prép1 # <E>
  - (N1 =: Qu Pind) ou (N1 =: Qu Psubj)
    - (Prép1 # à) et (Prép1 # de)
      - non (N1 = où) et non (N1 lieu de N0) et non (N0 et N1 V) ..... **35R_part**
    - non (Prép1 # à) **7**
    - non (Prép1 # de) **8**
  - non (N1 =: Qu Pind) et non (N1 =: Qu Psubj)
    - N1 =: V0-inf W
      - non (N1 =:si P ou si P) et non (N1 =: combien P) .... **1_part**
    - non (N1 =: V0-inf W)
      - (N0 =: Qu P) et [extrap]
        - non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj).... **5_part**
      - non (N0 =: Qu P) ou non [extrap]
        - N1 = où
          - N0 et N1 V
            - Prép1 =: avec .... **35S_part**
          - non (N0 et N1 V)
            - N1 lieu de destination de N0 .... **35LD**
            - (N1 lieu source de N0) ou (N1 lieu statique de N0)
              - N1 V de N0 .... **34L0**
              - non (N1 V de N0)
                - N1 lieu source de N0 .... **35LS**
                - N1 lieu statique de N0 .... **35ST**
            - non (N1 lieu de N0)
              - non (N1 V de N0) **35LR**
        - non (N1 = où) et non (N1 lieu de N0)
          - Prép1 # à+<E>
            - N0 et N1 V
              - Prép1 =: avec .... **35S_part**
            - non (N0 et N1 V) .... **35R_part**
          - non (Prép1 # à) **33**
- non (Prép1 # <E>)
  - (N1 =: Qu Pind) ou (N1 =: Qu Psubj) ou (N1 = :si P ou si P) ou (N1 =: combien P)
    - (N1 =: Qu Psubj) et (Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W)) .. **12**
    - non (Qu Psubj =: Qu Ni Vsubj W = (Ni) (de Vi-inf W))
      - N0 =: V1-inf W
        - (N0 =: Qu P) et (N1 =: Nhum) et non (N1 =: Nconc) et non (N1 =: V0-inf W) et non [extrap] .... **4_part**
      - non (N0 =: V1-inf W) **6**
  - non (N1 =: Qu Pind) et non (N1 =: Qu Psubj) et non (N1 = :si P ou si P) et non (N1 =: combien P) .... **voir Fig. G.5**

FIGURE G.4: Arbre de classement des verbes (4)

2 arguments (suite)

- N1 =: V0-inf W .... **1_part**
- non (N1 =: V0-inf W)
  - N0 =: Qu P
    - (N0 rendre N1 V-adj) ou (N0 rendre N1 plus V-adj)
      - (N1 =: Nhum) et non (N1 =: Nconc)
        - non [extrap] .... **4_part**
      - non (N1 =: Nhum) ou (N1 =: Nconc)
        - [passif par] .... **32RA_part**
    - non (N0 rendre N1 V-adj) et non (N0 rendre N1 plus V-adj)
      - [extrap] .... **5_part**
      - non [extrap]
        - (N1 =: Nhum) et non (N1 =: Nconc) .... **4_part**
  - non (N0 =: Qu P)
    - [passif par]
      - thème N1
        - (N0 lieu source de N1) ou (N0 lieu de destination de N1) ou (N0 lieu statique de N1) .... **38L0_part**
        - (N1 lieu source de N0) ou (N1 lieu de destination de N0) ou (N1 lieu statique de N0) ou (N1 lieu de passage de N0) **38L1_part**
        - non (N0 lieu de N1) et non (N1 lieu de N0) **32R3**
      - non (thème N1)
        - N1 apparition **32A**
        - N1 disparition **32D**
        - non (N1 apparition) et non (N1 disparition)
          - N1 =: Npl obl **32PL**
          - non (N1 =: Npl obl) .... **voir Fig. G.6**
    - non [passif par]
      - (N0 lieu source de N1) ou (N0 lieu de destination de N1) ou (N0 lieu statique de N1)
        - N1 =: N-hum .... **38L0_part**
      - (N1 lieu source de N0) ou (N1 lieu de destination de N0) ou (N1 lieu statique de N0) ou (N1 lieu de passage de N0)
        - N1 =: N-hum .... **38L1_part**
      - non (N0 lieu de N1) et non (N1 lieu de N0) **32NM**

FIGURE G.5: Arbre de classement des verbes (5)

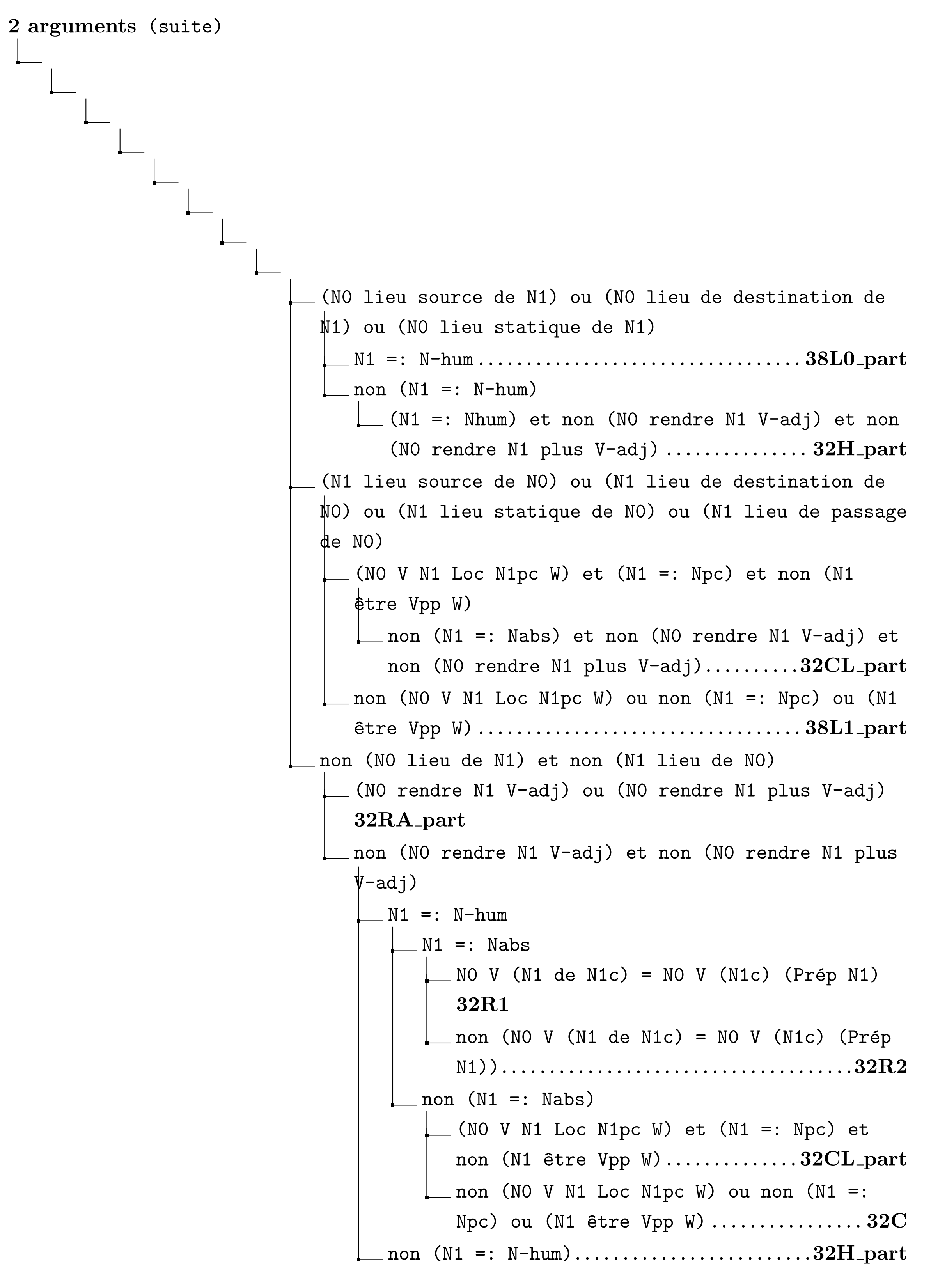


FIGURE G.6: Arbre de classement des verbes (6)

1 argument
- N0 =: N-hum.......... **31R**
- non (N0 =: N-hum) et N0 =: Nhum.......... **31H**

FIGURE G.7: Arbre de classement des verbes (7)